

ORSAY

Série A

N° d'ordre :

118

**T H E S E S**

PRESENTEES

**A LA FACULTE DES SCIENCES D'ORSAY  
UNIVERSITE DE PARIS**

POUR OBTENIR

**LE TITRE DE DOCTEUR-INGENIEUR**

par

**KHAN Solaiman**

**ETUDE ET REALISATION D'OPERATEURS D'ENCHAINEMENT**

Soutenues le 18 septembre 1970, devant la Commission d'Examen

**MM. BERTEIN**

Président

**DONADIEU  
DEBRAINE**

Examineurs



ORSAY  
Série A  
N° d'ordre :  
118

**T H E S E S**

**PRESENTEES**

**A LA FACULTE DES SCIENCES D'ORSAY  
UNIVERSITE DE PARIS**

**POUR OBTENIR**

**LE TITRE DE DOCTEUR-INGENIEUR**

**par**

**KHAN Solaiman**

**ETUDE ET REALISATION D'OPERATEURS D'ENCHAINEMENT**

Soutenues le 18 septembre 1970, devant la Commission d'Examen

**MM. BERTEIN**

Président

**DONADIEU  
DEBRAINE**

Examineurs



Je tiens à exprimer mes remerciements à M. DOIREAU, Chef du Département d'Electronique Générale et à M. POTTIER, Chef du Service d'Instrumentation Nucléaire, qui m'ont donné toutes facilités pour entreprendre et mener à bien ce travail de thèse. Je remercie également le Professeur P. GRIVET de la Faculté des Sciences d'Orsay, pour avoir bien voulu accepter mon inscription et un sujet d'avant-garde aussi particulier.

Je remercie également le Professeur BERTEIN d'avoir bien voulu accepter la présidence du jury de cette thèse ainsi que le Professeur DONADIEU membre de ce jury.

Je veux aussi exprimer ma gratitude au Professeur P. DEBRAINE, adjoint au Directeur du Centre d'Etudes Nucléaires de Saclay, Professeur d'informatique à l'Institut National des Sciences et Techniques Nucléaires, pour les cours qu'il m'a dispensés dans le domaine des techniques binaires appliquées aux ordinateurs.

Je tiens à dire également l'importance de la dette que j'ai contractée envers M. R.L. VALLEE, ingénieur au Commissariat à l'Energie Atomique et Professeur d'Electronique Générale au C.N.A.M. (Centre associé de Saclay), pour la constante attention et l'aide patiente qu'il m'a apportées, particulièrement dans la rédaction de ce texte en langue française.

Je remercie M. VERGEZ, ingénieur au Commissariat à l'Energie Atomique, pour ses commentaires et suggestions sur les méthodes graphiques qu'il a contribué à développer et pour les conseils pratiques qu'il m'a dispensés.

Bien qu'arrivant en finale dans la liste de mes remerciements, ce ne sont pas les moindres que j'adresse au Commissariat à l'Energie Atomique qui s'est attaché à me fournir les moyens matériels nécessaires à la poursuite d'un stage qui, sans doute, restera pour moi l'un des plus fructueux dans ma formation d'ingénieur.



Digitized by the Internet Archive  
in 2026

[https://archive.org/details/IA41551501\\_0029](https://archive.org/details/IA41551501_0029)

## INTRODUCTION

L'évolution des mathématiques dans le sens de la généralisation peut être divisée en trois étapes essentielles, à savoir :

- une étape élémentaire qui correspond à l'arithmétique,
- une étape intermédiaire où l'algèbre domine,
- enfin, une étape où l'analyse, et non plus l'algèbre, joue le principal rôle dans l'étude et la résolution pratique des problèmes qui se posent.

L'analyse permet en effet d'obtenir des solutions satisfaisantes là où les formules algébriques n'ont plus cours. Il se présente en effet de nombreux cas où les formules et modèles algébriques que nous proposent les mathématiques modernes ne sont pas très adaptés aux problèmes réels et ne conduisent pas aux solutions désirées. Le seul recours reste alors celui de l'analyse qui, non seulement permet de saisir plus facilement les données techniques, mais réduit aussi considérablement l'effort mental et le temps consacré à l'étude des problèmes.

C'est dans cette optique que "l'analyse binaire" a été adoptée plutôt qu'une algèbre de Boole.

Car un jour viendra sans doute où le spécialiste en informatique prendra conscience de la nécessité d'abandonner certaines conceptions traditionnelles moins bien adaptées pour adopter les nouvelles méthodes de l'analyse binaire.

Des recherches ont été menées avec succès dans ce domaine, au service d'instrumentation nucléaire du Centre d'Etudes Nucléaires de Saclay. Citons en particulier le rapport n° 35-34, intitulé "Analyse binaire", ouvrage de R.L. VALLEE, ingénieur au Commissariat à l'Energie Atomique, Département d'Electronique Générale, Service d'Instrumentation Nucléaire et professeur d'Electronique générale au C.N.A.M. (centre associé de Saclay). Cet ouvrage introduit avec toute la rigueur mathématique nécessaire, les bases théoriques essentielles de "l'analyse binaire" et son utilisation pour le calcul de systèmes concrets.

Le présent travail s'inscrit dans le cadre des applications qui résultent des théories ainsi développées et les idées de base de la thèse envisagée ont été empruntées à la publication du même auteur, intitulée "De l'intérêt d'une organisation locale des structures dans les automatismes numériques", article paru dans le bulletin d'instrumentation nucléaire BIN n° 36 de

mars 1969.

Les opérateurs d'enchaînement constituent la classe élémentaire des opérateurs programmes destinés à permettre la réalisation, sous la forme d'un standard modulaire, de systèmes automatiques complexes qui consacreront la décentralisation du contrôle avec séparation nette entre traitement de l'information et gestion par programme de ce traitement. L'opérateur d'enchaînement qui a été étudié et réalisé n'est qu'une première étape dont le succès constitue une promesse pour l'avenir des méthodes de l'analyse binaire appliquées aux systèmes de l'informatique.

# S O M M A I R E

## INTRODUCTION

Pages

|                                                                                                |    |
|------------------------------------------------------------------------------------------------|----|
| CHAPITRE I : Rappel des notions d'analyse binaire .....                                        | 1  |
| 1.1 Ensembles binaires algébriques .....                                                       | 1  |
| 1.2 Produits et produels .....                                                                 | 2  |
| 1.3 Propriétés particulières des produits et produels .....                                    | 2  |
| 1.4 Fonctions carrées biformes .....                                                           | 4  |
| 1.4.1 Théorème .....                                                                           | 4  |
| 1.5 Réalisations pratiques des fonctions binaires à l'aide de circuits logiques .              | 5  |
| CHAPITRE II : Rappel des méthodes d'étude générale des systèmes séquentiels .....              | 9  |
| 2.1 Généralités .....                                                                          | 9  |
| 2.2 Fonctions de base .....                                                                    | 10 |
| 2.3 Fonctions réflexes élémentaires .....                                                      | 12 |
| 2.4 Etude graphique des systèmes séquentiels .....                                             | 16 |
| 2.4.1 Séparation des fonctions .....                                                           | 17 |
| 2.5 Recherche et élimination des aléas de commutation .....                                    | 19 |
| 2.6 Etude de l'échantillonneur aléatoire .....                                                 | 20 |
| CHAPITRE III : Organisation actuelle des calculateurs électroniques .....                      | 27 |
| 3.1 Constitution du "software" .....                                                           | 27 |
| 3.2 Constitution du "hardware" .....                                                           | 28 |
| 3.3 Contrôle synchrone et asynchrone .....                                                     | 29 |
| 3.3.1 Les raisons d'un contrôle asynchrone .....                                               | 29 |
| 3.3.2 Systèmes hybrides .....                                                                  | 30 |
| 3.4 Développement historique .....                                                             | 30 |
| CHAPITRE IV : Les opérateurs d'enchaînement, forme élémentaire des opérateurs programmes ..... | 35 |
| 4.1 Généralités .....                                                                          | 35 |
| 4.2 Calcul d'un opérateur d'enchaînement .....                                                 | 37 |

|                                                                               |    |
|-------------------------------------------------------------------------------|----|
| 4.3 Aspect statistique des opérateurs d'enchaînement .....                    | 46 |
| CHAPITRE V : Réalisation pratique des opérateurs d'enchaînement .....         | 49 |
| 5.1 Composants utilisés .....                                                 | 49 |
| 5.2 Principales mesures effectuées sur les opérateurs d'enchaînement réalisés | 51 |
| 5.3 Montage expérimental de simulation .....                                  | 51 |
| CONCLUSION .....                                                              | 69 |

## CHAPITRE I

## RAPPEL DES NOTIONS D'ANALYSE BINAIRE

Nous allons rappeler rapidement dans ce premier chapitre, les notions fondamentales d'analyse binaire dont la terminologie, les définitions et les symboles sont maintenant bien établis et ont fait l'objet d'une étude détaillée dans le rapport CEA n° 35-34 et l'ouvrage intitulé "Analyse Binaire" dû à R.L. Vallée.

1.1 - Ensembles binaires algébriques

Si  $a$  et  $b$ ,  $a \neq b$  sont deux nombres appartenant à l'ensemble des nombres algébriques, les ensembles binaires algébriques " $E_{ab}$ " sont définis en partant des implications suivantes:

$$\bigcap \varphi_{ab} \in E_{ab}, \text{ lorsque } (\varphi_{ab} \neq a) \Rightarrow (\varphi_{ab} = b)$$

$$\text{et } (\varphi_{ab} \neq b) \Rightarrow (\varphi_{ab} = a)$$

une opération  $*$ , définie dans l'ensemble des nombres algébriques, qui admet  $a$  comme élément absorbant et  $b$  comme élément neutre, fournit un résultat qui appartient à l'ensemble  $E_{ab}$ , à condition que les termes qui interviennent dans cette opération appartiennent également à l'ensemble  $E_{ab}$ .

$$\bigvee (\varphi_1 \varphi_2) \in E_{a,b} \Rightarrow \varphi_1 * \varphi_2 \in E_{a,b}$$

Nous pouvons écrire en effet :

$$\bigvee \varphi, a * \varphi = \varphi * a = a \text{ (élément absorbant "a")}$$

$$\text{et } b * \varphi = \varphi * b = \varphi \text{ (élément neutre "b")}$$

Le produit algébrique, qui admet pour élément neutre "1" et pour élément absorbant "0" appartient ainsi à l'ensemble " $E_{01}$ " lorsque chaque facteur appartient aussi à  $E_{01}$ . Cela justifie le choix qui a été fait de cet ensemble pour l'étude des systèmes logiques.

Tout ensemble binaire algébrique " $E_{ab}$ " admet une relation d'automorphisme qui s'écrit d'une façon générale :

$$\boxed{\varphi_{ab} + \bar{\varphi}_{ab} - (a + b) = 0} \quad (1)$$

Cette relation appelée, par définition, dualité, correspond, dans le cas particulier de l'ensemble " $E_{01}$ " à l'expression :

$$\varphi + \bar{\varphi} - 1 = 0, \text{ soit } \bar{\varphi} = 1 - \varphi$$

Ceci mène à la relation d'involution :

$$\bar{\bar{\varphi}} = 1 - \bar{\varphi} = \varphi, \text{ et plus généralement aux relations de De Morgan} \quad (2)$$

### 1.2 - Produits et Produels

Considérons les éléments, variables ou fonctions binaires

$(\varphi_1, \varphi_2, \dots, \varphi_n) \in E_{01}$ , nous avons vu que le produit de ces éléments appartient également à l'ensemble " $E_{01}$ " :

$$P = \varphi_1 \times \varphi_2 \times \dots \times \varphi_n \in E_{01}.$$

En permutant l'élément neutre et l'élément absorbant, nous pouvons définir, par dualité, une opération  $\eta$  qui appartient aussi à l'ensemble  $E_{01}$  et que nous appellerons "produel".

Cette relation  $\eta$  peut s'écrire algébriquement :

$$\eta = 1 - (1 - \varphi_1)(1 - \varphi_2) \dots (1 - \varphi_n).$$

Elle est associative et commutative et il existe entre produel et produit une propriété réciproque de distributivité qui induit une représentation symbolique du produel suivant une verticale \* :

$$\eta = 1 - (1 - \varphi_1)(1 - \varphi_2) \dots (1 - \varphi_n) = \overline{\varphi_1 \cdot \varphi_2 \dots \varphi_n} \quad (4)$$

$\left| \begin{array}{c} \varphi_1 \\ \varphi_2 \\ \cdot \\ \cdot \\ \cdot \\ \cdot \\ \varphi_n \end{array} \right|$

$\varphi_1 \cdot \varphi_2 \dots \varphi_n$  sont appelés facteurs duals, par analogie avec les facteurs d'un produit.

### 1.3 - Propriétés particulières des produits et produels

Toutes les propriétés des produits et des produels sont des conséquences démontrables des définitions algébriques qui viennent d'être données. Tous les axiomes de "l'algèbre de Boole" correspondent, en analyse binaire, à des théorèmes rigoureusement démontrés.

Théorème d'idempotence :

$$\overbrace{0^n = 0 \text{ et } 1^n = 1}^n, \text{ nous en déduisons : } \overbrace{x, x, x, x, x, \dots, x, x}^n \quad x^n = x \text{ lorsque } x \in E_{01}$$

De même  $\left| \begin{array}{c} x \\ x \\ x \\ . \\ . \\ . \\ x \end{array} \right|_n = 1 - (1 - x)^n = x$

Conséquences :

$$\left| \frac{x}{x} \right| = 1 - x(1 - x) = 1 - x + x^2 = 1 - x + x = 1 ,$$

$$x \cdot \bar{x} = x(1 - x) = x - x^2 = x - x = 0$$

$$\left| \frac{1}{x} \right| = 1 - (1 - 1)(1 - x) = 1 - 0 = 1$$

Corollaires :

$$\bigvee_{(A, B) \in E_{01}}$$

$$\text{i)} \left| \begin{array}{c} A \cdot B \\ \bar{A} \cdot B \end{array} \right| = \left| \begin{array}{c} A \\ \bar{A} \end{array} \right| B = B , \quad \text{ii)} \left| \begin{array}{c} A \\ B \end{array} \right| \left| \begin{array}{c} \bar{A} \\ B \end{array} \right| = \left| \begin{array}{c} A \cdot \bar{A} \\ B \end{array} \right| = B$$

$$\text{iii)} \quad A \left| \begin{array}{c} A \\ B \end{array} \right| = \left| \begin{array}{c} A \\ 0 \end{array} \right| \left| \begin{array}{c} A \\ B \end{array} \right| = \left| \begin{array}{c} A \\ 0 \cdot B \end{array} \right| = A$$

$$\text{iv)} \quad \left| \begin{array}{c} A \\ A \cdot B \end{array} \right| = \left| \begin{array}{c} A \times 1 \\ A \times B \end{array} \right| = A \left| \begin{array}{c} 1 \\ B \end{array} \right| = A$$

Les produits et les produels étant réciproquement distributifs, nous pouvons procéder à des mises en facteur, ou en facteur dual, des termes binaires. Mais nous pouvons également effectuer les produits ou les produels dans les expressions symboliques. Nous donnons, à titre d'exemple, quelques opérations "type" en les comparant à celles qui leur seraient équivalentes en algèbre de Boole :

#### Analyse binaire

mise en facteur :

$$\left| \begin{array}{c} a \cdot b \\ a \cdot c \\ a \cdot d \end{array} \right| = a \cdot \left| \begin{array}{c} b \\ c \\ d \end{array} \right| \quad (a \cdot b)V(a \cdot c)V(a \cdot d) = a(bVcVd)$$

mise en facteur dual :

$$\left| \begin{array}{c} a \\ b \end{array} \right| \left| \begin{array}{c} a \\ c \\ d \end{array} \right| = \left| \begin{array}{c} a \\ b \cdot c \cdot d \end{array} \right| \quad (aVb) \cdot (aVc) \cdot (aVd) = aV(b \cdot c \cdot d)$$

Suppression du trait médian d'une fonction carrée biforme :

$$\left| \begin{array}{c} x \\ a \end{array} \right| \left| \begin{array}{c} b \\ \bar{x} \end{array} \right| = \left| \begin{array}{c} x \cdot b \\ a \cdot \bar{x} \end{array} \right| \quad (xVa) \cdot (bV\bar{x}) = xbVab \vee a\bar{x} = xb \vee a\bar{x}$$

#### Algèbre de Boole

Développement par produits effectués :

$$\left| \begin{array}{c|c} a_1 & b_1 \\ \hline a_2 & b_2 \end{array} \right| = \left| \begin{array}{c} a_1 b_1 \\ a_2 b_1 \\ a_1 b_2 \\ a_2 b_2 \end{array} \right| \quad \begin{aligned} & (a_1 V a_2) \cdot (b_1 V b_2) \\ & = a_1 b_1 V a_1 b_2 V a_2 b_1 V a_2 b_2 \end{aligned}$$

Développement par produits effectués :  $a_1 a_2 V b_1 b_2$

$$\left| \begin{array}{c|c} a_1 \cdot a_2 \\ \hline b_1 \cdot b_2 \end{array} \right| = \left| \begin{array}{c|c|c|c} a_1 & a_1 & a_2 & a_2 \\ \hline b_1 & b_2 & b_1 & b_2 \end{array} \right| \quad \begin{aligned} & = (a_1 V b_1) \cdot (a_1 V b_2) \\ & (a_2 V b_1) \cdot (a_2 V b_2) \end{aligned}$$

#### 1.4 - Fonctions carrées biformes

Une fonction carrée biforme est, par définition, un produit de produits ou un produit de deux produits comprenant, au total, quatre termes binaires disposés en carré dont deux, au moins, sont complémentaires.

$$\left| \begin{array}{c|c} A \cdot \varphi \\ \hline \bar{\varphi} \cdot B \end{array} \right| \text{ ou } \left| \begin{array}{c|c} A & \varphi \\ \hline \bar{\varphi} & B \end{array} \right|$$

Dans l'étude des simplifications des fonctions binaires, les fonctions carrées biformes jouent un rôle fondamental. L'importance de ces fonctions résulte du théorème suivant :

**1.4.1 - Théorème :** Une fonction carrée biforme n'est pas modifiée par la suppression du tracé vertical médian, à condition de disposer les facteurs complémentaires suivant une diagonale du carré correspondant à son expression symbolique.

Ce théorème est de loin le plus utilisé en analyse binaire. Il ne peut pas être énoncé en algèbre de Boole où il conduit malheureusement à la règle suivante qui entraîne un réarrangement complet des symboles et ne présente, par suite, que très peu d'intérêt :

$$(A \cdot \varphi) V (\bar{\varphi} \cdot B) = (A V \bar{\varphi}) \cdot (B V \varphi)$$

Cela est d'autant plus regrettable que toute fonction peut s'exprimer par une fonction carrée biforme :

En effet :

$$\begin{aligned} \text{i) } F_1 &= \left| \begin{array}{c|c} A \cdot \varphi \\ \hline \cdot & B \end{array} \right| = \left| \begin{array}{c|c} A \cdot \varphi \\ \hline B & \varphi \end{array} \right| = \left| \begin{array}{c|c} A & \varphi \\ \hline B & \bar{\varphi} \cdot B \end{array} \right| \\ \text{ii) } F_2 &= A \left| \begin{array}{c|c} B \\ \hline \varphi \end{array} \right| = \left| \begin{array}{c|c} A & B \\ \hline 0 & \varphi \end{array} \right| = \left| \begin{array}{c|c} A & B \\ \hline \varphi \cdot \bar{\varphi} & \varphi \end{array} \right| = \left| \begin{array}{c|c} A \cdot B & \bar{\varphi} \\ \hline \varphi & A \end{array} \right| \\ \text{iii) } F_3 &= \left| \begin{array}{c|c} A \cdot \varphi \\ \hline \bar{\varphi} \cdot B \\ \hline C \end{array} \right| = \left| \begin{array}{c|c} A \cdot \varphi \\ \hline \bar{\varphi} \cdot B \\ \hline C & \varphi \end{array} \right| = \left| \begin{array}{c|c} A & \varphi \\ \hline C & \bar{\varphi} \\ \hline & B \\ & C \end{array} \right| \end{aligned}$$

$$\text{iv) } F_4 = \left| \begin{array}{c|c} A & \bar{\varphi} \\ \hline \varphi & B \end{array} \right| \cdot C = \left| \begin{array}{c|c|c} A & \bar{\varphi} & C \\ \hline \varphi & B & \varphi \cdot \bar{\varphi} \end{array} \right| = \left| \begin{array}{c|c} \varphi & B \cdot C \\ \hline A \cdot C & \bar{\varphi} \end{array} \right|$$

Notons que les fonctions carrées biformes permettent, très simplement de définir les consensus direct et dual qui correspondent respectivement au produit et au produit des termes diagonaux non complémentaires.

Soit la fonction :

$$F = \left| \begin{array}{c|c} A \cdot \varphi & \\ \hline \bar{\varphi} \cdot B & \end{array} \right| = \left| \begin{array}{c|c} A & \varphi \\ \hline \bar{\varphi} & B \end{array} \right|$$

- le consensus direct "A . B" est facteur dual de F :

$$F = \left| \begin{array}{c|c} A \cdot \varphi & \\ \hline \bar{\varphi} \cdot B & \end{array} \right| \quad , \quad \text{et le consensus dual} \quad \left| \begin{array}{c} A \\ B \end{array} \right|$$

est facteur de F :

$$F = \left| \begin{array}{c|c|c} A & \varphi & A \\ \hline \bar{\varphi} & B & B \end{array} \right|$$

Les consensus permettent dans la plupart des cas la simplification optimale des fonctions combinatoires mais ils sont utiles surtout pour la résolution des aléas de commutation comme nous le verrons au cours de l'étude des opérateurs d'enchaînement.

#### 1.5 - Réalisation pratique des fonctions binaires à l'aide de circuits logiques

Rappelons que la technologie actuelle des circuits intégrés met généralement à notre disposition des portes NI (NOR) et des portes "ON" (NAND).

Nous représentons ces portes par les symboles suivants (fig. 1) :

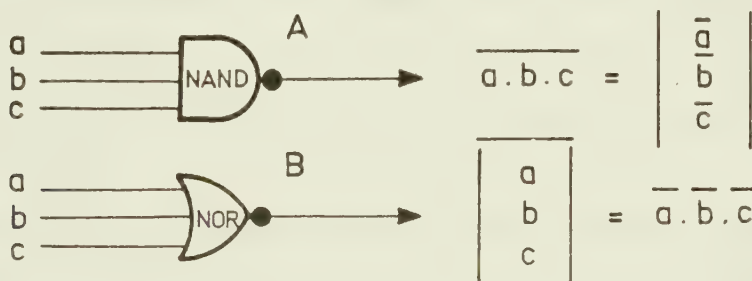


Figure 1 - A) Porte NAND

B) Porte NOR

Pour réaliser en deux couches une fonction binaire donnée, nous savons : qu'il faut l'écrire sous forme d'un produit de produits si l'on utilise des circuits NI et sous forme d'un produit de produits si l'on utilise des circuits "NAND".

Exemple :

$$f = \begin{vmatrix} a & \bar{b} \\ b & \bar{c} \\ c & a \end{vmatrix} = \begin{vmatrix} a & \bar{a} \\ b & \bar{b} \\ c & \bar{c} \end{vmatrix}$$

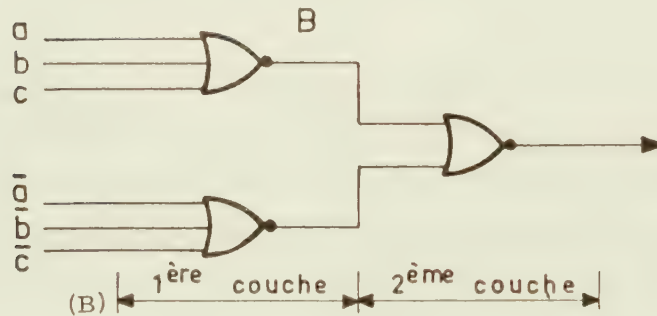
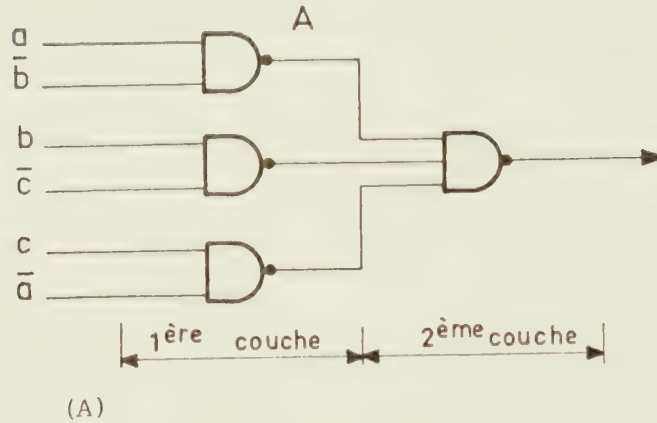


Figure 2 - A) Circuit réalisé en deux couches à l'aide de portes NAND  
B) Circuit réalisé en deux couches à l'aide de portes NOR

A condition de disposer d'un nombre suffisant d'entrées sur chaque porte élémentaire, nous pouvons toujours réaliser une fonction binaire en trois couches au maximum. Reprenons les expressions précédentes en supposant que l'on ne dispose que des variables directes  $a$ ,  $b$  et  $c$ , il est possible, en ajoutant trois portes de complément, d'obtenir la fonction en trois couches (fig. 3)

Grâce aux méthodes précises de l'analyse binaire, tous les problèmes des circuits combinatoires ou séquentiels que nous avons rencontrés dans l'étude des opérateurs d'enchaînement ont pu être résolus sans faire appel à la méthode traditionnelle.

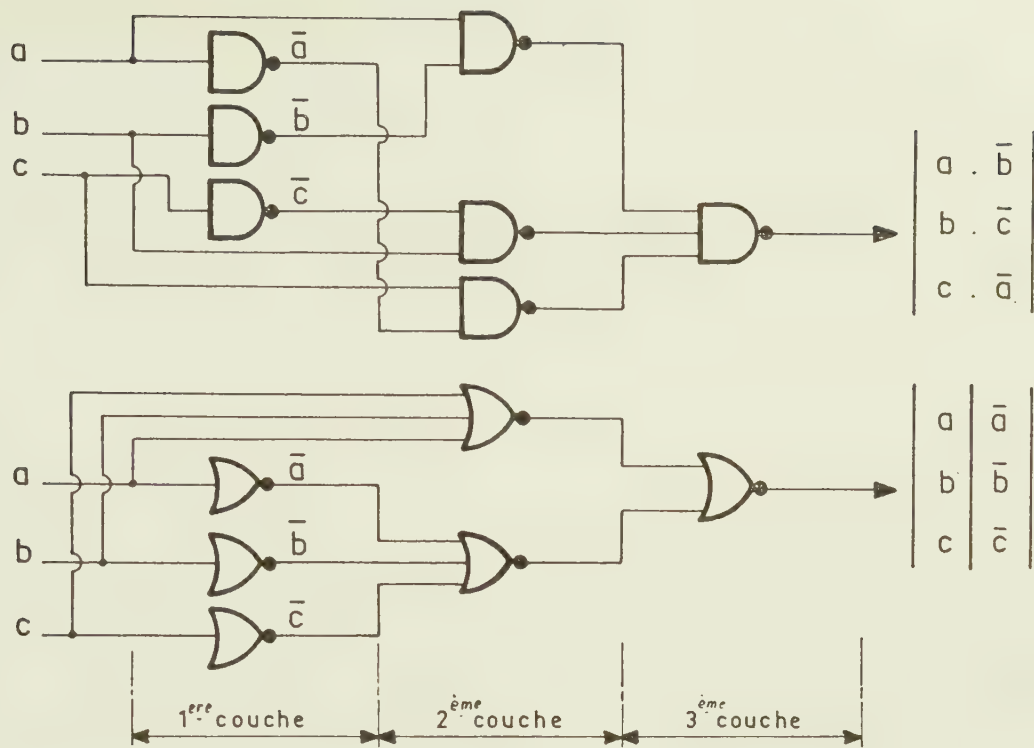


Figure 3



## CHAPITRE II

RAPPEL DES METHODES D'ETUDE GENERALE  
DES SYSTEMES SEQUENTIELS

## 2.1 - Généralités

L'absence d'élément symétrique dans les opérations de base de l'analyse binaire conduit à faire une distinction précise entre variables et fonctions binaires.

Nous désignerons donc le vecteur fonction d'un système par le nombre, exprimé en numération binaire, obtenu en associant dans un ordre donné, choisi à l'avance, les valeurs binaires 0 ou 1 prises par chacune des fonctions  $y_i \in E_{01}$  qui composent le vecteur.

$$Y_p = y_p, y_{p-1}, y_{p-2}, \dots, y_i, \dots, y_2, y_1 \quad y_i \in E_{01}$$

Nous appellerons également, vecteur variable, le nombre associé aux variables binaires rangées dans un ordre donné choisi à l'avance.

$$X_n = x_n, x_{n-1}, \dots, x_j, \dots, x_2, x_1, \quad x_j \in E_{01}$$

L'état du système est alors défini par le nombre associé à l'ensemble des valeurs correspondantes des deux vecteurs, variable et fonction, groupés dans l'ordre  $X_n, Y_p$ .

L'état,  $\xi_{\infty} = X_n^{\circ}, Y_p^{\circ}$ , correspond, par exemple, au cas où le vecteur fonction du système  $Y_p$  prend la valeur  $Y_p^{\circ}$  alors que  $X_n$  est égal à  $X_n^{\circ}$ .

On distingue habituellement, les états stables où le vecteur fonction conserve sa valeur aussi longtemps que le vecteur variable conserve également sa propre valeur, et les états de commutation, qui sont des états fugitifs dépendant de la rapidité de réponse des circuits, où le vecteur fonction conserve encore sa valeur pendant un court instant alors que le vecteur variable vient d'être modifié entraînant une ou plusieurs commutations successives. Entre deux états stables consécutifs il existe, au moins, un état de commutation, lorsque  $Y_p$  varie.

Les états d'un système peuvent être représentés, de plusieurs façons différentes sur un graphe cartésien ramené aux axes de coordonnées discontinues  $OX_n, OY_p$ .

Le fonctionnement du système peut être entièrement décrit par ces graphes si l'on y associe une représentation sagittale des commutations résultant de valeurs adjacentes du vecteur variable. Il est exclu, cependant, d'envisager une commutation entre deux états stables situés sur une même verticale. Le vecteur fonction ne peut, en effet, changer spontanément de

valeur sans que le vecteur variable n'ait préalablement changé de valeur.

Enfin l'étude générale des systèmes numériques conduit à considérer trois classes de fonctions essentielles qui sont :

- les fonctions de transcodage, les fonctions génératrices et les fonctions réflexes.

On démontre [3] qu'un système séquentiel peut toujours se décomposer en groupes de fonctions qui appartiennent nécessairement à l'une ou l'autre de ces trois classes.

## 2.2 - Fonctions de base

Les fonctions de transcodage se déduisent directement des fonctions binaires combinatoires, lesquelles constituent les composantes du vecteur fonction. Le qualificatif "transcodage" rappelle l'usage le plus courant qu'il est fait de ces fonctions pour transformer les codes des informations numériques selon les traitements prévus ou les buts envisagés.

Les fonctions de transcodage peuvent revêtir quatre formes principales qui sont :

- la forme développée, la forme maillée ou arborescente, la forme paramétrique et la forme itérative. Dans tous les cas, elles font correspondre à chaque valeur du vecteur variable, une valeur et une seule, du vecteur fonction associé.

Cette particularité univoque caractérise les fonctions de transcodage qui traduisent, sous forme de relations binaires, une injection de l'ensemble des valeurs du vecteur variable, dans l'ensemble des valeurs prises par le vecteur fonction. Une fonction de transcodage  $Y_p = F(X_n)$  est une fonction algébrique explicite discontinue comprenant un nombre fini et limité de points.

Le transcodage est optimal lorsque  $n = p$ .

Les fonctions génératrices sont des fonctions dont les composantes n'ont plus de représentation combinatoire possible parcequ'elles engendrent des transitions de valeurs binaires. Les transitions correspondent à des intervalles de commutation que nous écrivons pour une variable binaire " $x$ " :

-  $x = (0 \rightarrow 1)$ , " $x$ " égale à transition zéro-un,

-  $x = (1 \rightarrow 0)$ , " $x$ " égale à transition un-zéro,

ou  $x = (0 \rightarrow 1 \rightarrow 0)$ , " $x$ " égale à la double transition zéro-un-zéro,

ou encore,  $x = (1 \rightarrow 0 \rightarrow 1)$ , " $x$ " égale à la double transition un-zéro-un.

Lorsque les valeurs " $0$ " et " $1$ " sont effectivement atteintes, lors des doubles transitions, nous pouvons écrire également :

$x = (0 \rightarrow 1, 1 \rightarrow 0)$  ou  $x = (1 \rightarrow 0, 0 \rightarrow 1)$

La transition est un concept qui relève de l'analyse séquentielle.

Elle n'appartient pas à l'ensemble " $E_0$ " et n'admet aucune définition algébrique.

Les fonctions génératrices les plus souvent utilisées sont la fonction "délai" et la fonction "impulsion".

On appelle fonction "délai", et l'on représente symboliquement par " $\Delta_{\tau}(x)$ ", toute fonction binaire de la variable binaire " $x$ ",  $[\Delta_{\tau}(x), x] \in E_{01}$  qui, avec un retard constant égal à " $\tau$ ", prend les mêmes valeurs que la variable " $x$ " dont elle dépend.

Dans certains cas, les retards peuvent être différents pour les transitions  $(0 \rightarrow 1)$ , et  $(1 \rightarrow 0)$  de la variable " $x$ ". Si " $\tau_0$ " et " $\tau_1$ " sont les retards qui correspondent respectivement à ces deux transitions, nous pouvons écrire symboliquement la fonction délai considérée :

$$\Delta_{\tau_0, \tau_1}(x). \text{ Nous avons ainsi } \Delta_{\tau}(x) = \Delta_{\tau, \tau}(x).$$

Remarquons que les fonctions "délais" satisfont aux égalités binaires suivantes :

$$\overline{\Delta_{\tau}(x)} = \Delta_{\tau}(\bar{x})$$

$$\Delta_{\tau}(x) = \overline{\Delta_{\tau}(\bar{x})}$$

On appelle fonction "impulsion" de la variable " $x$ " ou "impulsion de  $x$ ", la fonction " $I(x)$ " égale à "0" quelle que soit la valeur binaire de " $x$ " mais assurant la double transition  $(0 \rightarrow 1, 1 \rightarrow 0)$  pour  $x = (0 \rightarrow 1)$ . On admet, par définition que la première transition  $(0 \rightarrow 1)$  de  $I(x)$  coïncide avec la transition  $(0 \rightarrow 1)$  de " $x$ " :  $x = (0 \rightarrow 1) = I(x)$ .

Cette définition est illustrée par la table de vérité et le diagramme de la figure

2,21 :

| x | I(x) |
|---|------|
| 0 | 0    |
| 0 | 0    |
| ↓ | ↓    |
| 1 | 1    |
| 1 | 1    |
| ↓ | ↓    |
| 1 | 0    |
| 0 | 0    |

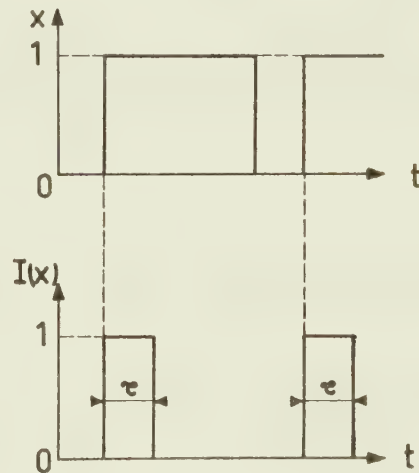


Figure 2,21 - Table de vérité et diagramme de la fonction impulsion

On peut, de la même manière, définir les fonctions impulsions  $\overline{I(x)}$ ,  $I(\bar{x})$  et  $\overline{I(\bar{x})}$ , obtenues en complétant la fonction, la variable, ou les deux simultanément. Les fonctions "impulsions" ne sont pas susceptibles d'une représentation algébrique, mais peuvent entrer dans des relations binaires :

$$x \cdot I(x) \equiv I(x)$$

$$I_{\tau}(x) = x \cdot \overline{\Delta_{\tau}(x)}$$

$$\left| \begin{array}{c} x \\ I(x) \end{array} \right| \equiv x$$

$$\Delta_{\tau}(x) = \begin{vmatrix} x & \overline{I_{\tau}(x)} \\ I_{\tau}(x) & \end{vmatrix}$$

Les fonctions réflexes, quant à elles, ont fait l'objet d'une étude très détaillée, parcequ'elles constituent la structure essentielle et fondamentale de tous les systèmes séquentiels. Ce sont des fonctions que l'on peut mettre sous une forme semi-implicite :  $Y_p = \Phi(X_n, Y_p)$ , et qui possèdent la propriété spécifique de pouvoir faire correspondre plusieurs valeurs du vecteur fonction " $Y_p$ " à une même valeur du vecteur variable " $X_n$ ".

On peut envisager une représentation graphique de ces fonctions dans un système d'axes  $OX_n, OY_p$ .

Le graphe séquentiel obtenu est discontinu. Il est fini et comprend, en général, un certain nombre de points stables reliés entre eux dans un ordre de succession qui est orienté. L'orientation dépend de la fonction et le graphe la caractérise pratiquement.

Lorsqu'une variation de " $X_n$ " modifie la fonction " $Y_p$ ", cette dernière prend une valeur finale qui dépend également de sa propre valeur avant la modification.

Mais il est exclu que le vecteur fonction puisse changer de valeur spontanément si le vecteur de commande n'a subi aucune variation. Cela résulte de la distinction fondamentale qui doit nécessairement être faite entre variables et fonctions.

Les états d'un système ayant été définis,  $\mathcal{E}_{kl} = X_n^k, Y_p^l$ , leur nombre étant fini, il faut pour qu'un système soit possible, que le nombre total d'états " $N_t$ " d'une part et le nombre maximal d'états " $N_o$ " pour la valeur donnée de  $X_n = X_n^o$  d'autre part, vérifient la condition des états :

$$N_t \leq 2^{n+p} \quad \text{et} \quad N_o \leq 2^p.$$

Cette condition des états s'applique aux états stables seulement lorsqu'ont été définies les fonctions élémentaires "mémoires" et "dibinaires" qui constituent le système.

### 2.3 - Fonctions réflexes élémentaires

Les fonctions réflexes élémentaires sont obtenues lorsque le vecteur fonction " $Y_p$ " se réduit à une seule composante binaire  $Y_p = y \in E_{01}$ .

Les fonctions mémoires élémentaires,  $y = \varphi(X_n, y)$ , peuvent se présenter, sous les trois formes possibles suivantes :

- fonction mémoire à remise à zéro prioritaire ou "R" prioritaire,

$$y = \begin{vmatrix} y \\ S \end{vmatrix} \bar{R}$$

Elle correspond au graphe de la figure 2.31.

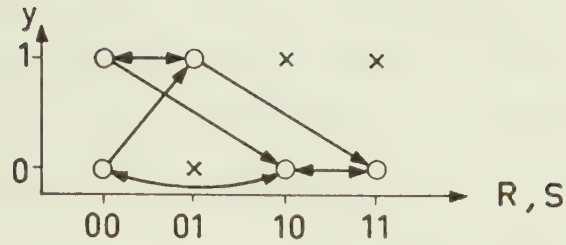


Figure 2.31 - Graphe de la fonction mémoire à "R" prioritaire.

- fonction mémoire à armement prioritaire ou "S" prioritaire,

$$y = \begin{vmatrix} y & \bar{R} \\ S \end{vmatrix}$$

Elle correspond au graphe de la figure 2.32.

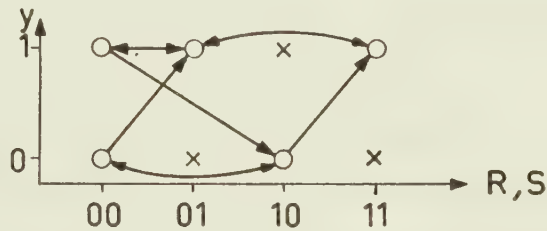


Figure 2.32 - Fonction mémoire à "S" prioritaire.

- fonction mémoire symétrique ou à R, S symétriques,

$$y = \begin{vmatrix} y & \bar{R} \\ S & \bar{R} \end{vmatrix} = \begin{vmatrix} y & \bar{R} \\ S & \bar{R} \end{vmatrix}$$

Elle correspond au graphe de la figure 2.33,

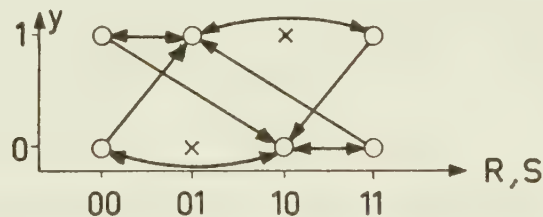


Figure 2.33 - Graphe de la fonction mémoire symétrique

"R et S" représentent, dans les différents cas, des fonctions du vecteur variable d'entrée " $X_n$ ",  $R(X_n)$  et  $S(X_n)$ .

Dans le cas où l'identité  $R.S \equiv 0$  est vérifiée, les trois formes précédentes sont confondues et d'une façon générale nous pouvons énoncer le théorème suivant qui permet de caractériser les fonctions "mémoires" élémentaires.

### - Théorème de caractérisation d'une fonction mémoire élémentaire

Pour qu'une fonction binaire  $y(X_n) \in E_{01}$ , qui dépend de "n" variables également binaires constituant le vecteur " $X_n$ ", soit une fonction "mémoire", il faut et il suffit qu'il existe trois groupes  $C_0$ ,  $C_1$  et  $C_2$  distincts de combinaisons de valeurs des variables, tels que toute combinaison du groupe " $C_0$ " entraîne nécessairement  $y = 0$ , toute combinaison du groupe " $C_1$ " entraîne nécessairement  $y = 1$ , et toute combinaison de " $C_2$ " n'ait aucune incidence sur la valeur de la fonction " $y$ ", ( $y = y$ ).

La démonstration de ce théorème est donnée dans l'ouvrage "d'analyse binaire" de René L. Vallée.

Les groupes " $C_0$ ", " $C_1$ " et " $C_2$ " font généralement partie des données immédiates d'un problème.

En désignant symboliquement par  $/C_0/$  et  $/C_1/$ , les produits de produits qui correspondent à  $/C_0/ = 1$  et  $/C_1/ = 1$  pour les combinaisons de valeurs des variables constituant respectivement les groupes " $C_0$ " et " $C_1$ ", les fonctions d'entrée  $R(X_n)$  et  $S(X_n)$  de la fonction mémoire associée sont données par les expressions  $R(X_n) = /C_0/$  et  $S(X_n) = /C_1/$ .

Nous en déduisons :

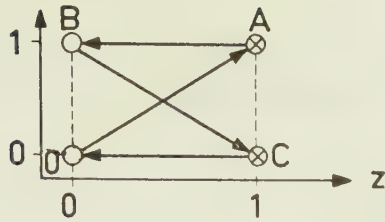
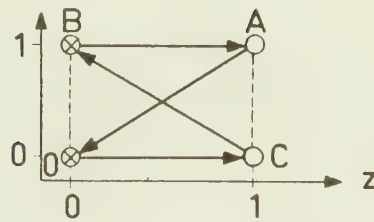
$$y = \left| \begin{array}{c} y \\ /C_1/ \end{array} \right| /C_0/ = \left| \begin{array}{c} y /C_0/ \\ /C_1/ \end{array} \right|$$

puisque les combinaisons des groupes " $C_0$ " et " $C_1$ " sont distinctes.

- Les fonctions "dibinaires" constituent une deuxième famille de fonctions réflexes élémentaires.

Considérons les fonctions réflexes  $Z = f(Z, z)$  dans lesquelles les vecteurs, fonction et variable, ne comprennent chacun, respectivement, qu'une seule composante binaire,  $(Z, z) \in E_{01}$ . Nous pouvons associer à ces fonctions, un graphe comprenant quatre points stables O, A, B, C, au plus, qui correspondent aux quatre états d'équilibre stable possibles ;  
O ( $z = 0, Z = 0$ ), A ( $z = 1, Z = 1$ ), B ( $z = 0, Z = 1$ ), C ( $z = 1, Z = 0$ ).

Les états correspondant aux points situés sur une même verticale, OB et CA, ne peuvent, en aucun cas, être des états stables consécutifs puisque les fonctions réflexes ne peuvent changer spontanément de valeur sans que le vecteur variable n'ait changé préalablement. Il existe alors une façon et une seule de joindre les points O, A, B, C qui consiste à suivre le contour d'un quadrilatère étoilé admettant ces points comme sommets. Il n'existe, sur ce quadrilatère, que deux sens de parcours seulement. Ces deux sens de parcours possibles mènent, en définitive, à deux fonctions particulières, entièrement définies par les graphes de la figure 2.34.

$Z = B(Z, z)$ 

 $Z = Bi(Z, z)$ 


2.34 a) - Graphe d'une fonction dibinaire directe

2.34 b) - Graphe d'une fonction dibinaire inverse.

Figure 2.34 - Graphes de définition des fonctions "dibinaires"

point d'équilibre stable

point de superposition d'un état d'équilibre stable et d'un état de commutation.

Rappelons que les fonctions "dibinaires" réalisées en pratique doivent satisfaire à la condition des seuils.

Pour qu'un circuit séquentiel élémentaire qui dépend d'une seule variable binaire puisse être identifié à une fonction dibinaire, il faut et il suffit qu'il présente quatre états d'équilibre stable consécutifs et distincts, qui conditionnent quatre seuils intermédiaires de commutation ramenés à l'entrée du circuit, séparant les quatre états stables, et distribués de telle sorte que les deux seuils franchis par les niveaux croissants du signal d'entrée soient supérieurs aux deux seuils franchis par les niveaux d'entrée décroissants.

Les fonctions dibinaires sont des diviseurs par deux et l'on peut établir, entre les deux formes directe et inverse, les relations suivantes :

$$B(Z, \bar{z}) = Bi(Z, z)$$

$$Bi(Z, \bar{z}) = B(Z, z)$$

$$B(\bar{Z}, z) = B(Z, z)$$

$$Bi(\bar{Z}, z) = Bi(Z, z).$$

Comme pour les fonctions "mémoires", il existe pour les fonctions "dibinaires" un théorème qui permet de les caractériser. La forme la plus générale d'une fonction dibinaire est la forme "R, S, z" que l'on écrit :

$$Q = B(Q, R, S, z) \quad \text{ou} \quad Q = Bi(Q, \bar{R}, \bar{S}, \bar{z}).$$

Théorème de caractérisation d'une fonction "dibinaire"

La condition nécessaire et suffisante pour qu'une fonction  $Q(X_n) \in E_{01}$  ("X<sub>n</sub>" vecteur variable) puisse être représentée par une fonction "dibinaire" est qu'il existe au moins un groupe "C<sub>3</sub>" de combinaisons, non consécutives, de valeurs des variables binaires  $x_n, x_{n-1}, \dots, x_2, x_1 = X_n$ , tel que l'une de ces combinaisons, lorsqu'elle apparaît assure la transition  $(Q \rightarrow \bar{Q})$ .

C'est-à-dire que si  $Q = 0$ , l'apparition de la combinaison

$$(X_n^0 \subset C_3) \text{ entraîne } Q = (0 \rightarrow 1), \text{ et si } Q = 1, \quad (X_n^1 \rightarrow X_n^0) \Rightarrow Q = (1 \rightarrow 0)$$

avec  $X_n^0 \subset C_3$ ,  $X_n^1 \not\subset C_3$ .

D'une façon générale, s'il existe une fonction binaire " $Q \in E_{01}$ " et quatre groupes distincts des valeurs du vecteur variable " $X_n$ " tels que l'on puisse écrire :

$$\begin{aligned} (X_n \subset "C_0") &\Rightarrow (Q = 0) \\ (X_n \subset "C_1") &\Rightarrow (Q = 1) \\ (X_n \subset "C_2") &\Rightarrow (Q = Q, \forall Q) \\ (X_n \subset "C_3") &\Rightarrow (Q \rightarrow \bar{Q}) \end{aligned}$$

$Q$  est alors une fonction dibinaire du type  $Q(R, S, z)$ .

Il faut dans le dernier cas, (groupe " $C_3$ ") que les combinaisons de valeurs de " $X_n$ " ne soient pas consécutives.

Nous pouvons alors calculer  $Q = B(Q, R, S, z)$

où  $Q = Bi(Q, \bar{R}, \bar{S}, \bar{z})$  dans lesquelles :

$$\begin{aligned} R &= \left| C_0 \right| \\ S &= \left| C_1 \right| \\ z &= \left| C_3 \right| \end{aligned}$$

Dans le cas où la fonction elle-même intervient dans la fonction d'entrée, ce qui est possible pour une fonction dibinaire, on obtient alors une fonction du type J-K qui admet plusieurs expressions parmi lesquelles les deux suivantes :

$$\begin{aligned} Q &= Bi(Q, \bar{z}) \quad \text{avec} \quad z = H. \quad \left| \begin{array}{c} Q \\ J \end{array} \right| \left| \begin{array}{c} K \\ \bar{Q} \end{array} \right| = H. \quad \left| \begin{array}{c} Q \cdot K \\ J \cdot \bar{Q} \end{array} \right| \\ \text{ou} \\ Q &= B(Q, z) \quad \text{avec} \quad z = H. \quad \left| \begin{array}{c} Q \\ J \end{array} \right| \left| \begin{array}{c} K \\ \bar{Q} \end{array} \right| = H. \quad \left| \begin{array}{c} Q \cdot K \\ J \cdot \bar{Q} \end{array} \right| \end{aligned}$$

## 2.4 - Etude graphique des systèmes séquentiels

Le comportement d'un système séquentiel peut entièrement être décrit par un graphe cartésien comprenant en abscisses les valeurs du vecteur variable et en ordonnées les valeurs du vecteur fonction.

Dans ce graphe apparaissent à la fois les états stables ainsi que les états de commutation et une représentation sagittale des transitions d'états complète la description du système.

Un état de commutation est repéré à l'horizontale de l'état stable d'origine et à la verticale de l'état stable final (figure 2.401).

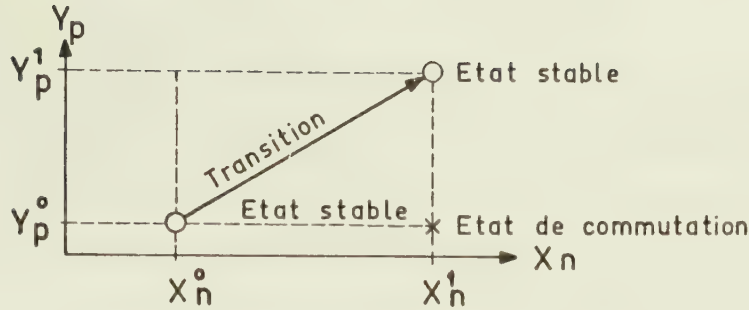


Fig. 2.401 - Représentation graphique adoptée pour la représentation du fonctionnement d'un système séquentiel.

Si nous savons reconnaître et utiliser des fonctions dibinaires quand la résolution d'un problème séquentiel l'exige, la condition des états se simplifie et nous pouvons nous contenter de considérer seulement les états stables " $N_{st}$ " du système.

Ce qui s'écrit :

$$N_{st} \leq 2^{n+p} \quad \text{avec} \quad N_{st} > 2^{n+p-1}.$$

avec, pour une valeur donnée de  $X_n = X_n^0$ ,

$$N_{X_n^0} \leq 2^p.$$

"p" représente alors le nombre total des fonctions réflexes élémentaires, mémoires et dibinaires.

#### 2.4.1 - Séparation des fonctions

Un système séquentiel peut toujours se décomposer selon les trois classes de fonctions qui sont :

- les fonctions de transcodage
- les fonctions réflexes
- les fonctions génératrices

La séparation des fonctions de transcodage se fait aisément sans modification de de la fonction séquentielle. Car l'information que fournit un transcodage est entièrement contenue dans les valeurs prises par le vecteur variable et ce dernier se trouve conservé, bien que transcodé, dans le graphe obtenu après réduction.

Les fonctions génératrices telles que horloge, oscillateurs, convertisseurs, etc... sont facilement repérables puisqu'elles font partie des données et doivent, compte tenu de leur particularité, être introduites comme composantes du vecteur variable de commande.

Les fonctions "délais" et "impulsions" font partie, pour leur part, des inconnues du problème et sont localisées et calculées en dernier lieu.

Il reste en définitive, les fonctions réflexes, qu'il faut repérer, puis séparer en fonctions "mémoires" et fonctions "dibinaires", sur un graphe réflexe réduit qui peut être établi à partir des données du problème séquentiel posé.

Les méthodes d'étude des graphes réflexes sont parfaitement établies et rigoureusement démontrées en "analyse binaire".

Partant du graphe réflexe réduit d'un système séquentiel il faut construire le graphe d'émergence de chaque fonction réflexe élémentaire en repérant pour cette fonction, les états commutants, les états commutés, et les états de commutation. Rappelons que l'on appelle, par définition, point commutant ou état commutant relatif à la composante binaire  $y_i \in E_{01}$  du vecteur fonction, tout point ou état stable, choisi comme tel, à condition qu'il précède immédiatement une transition de la fonction " $y_i \rightarrow \overline{y_i}$ ".

Un point commuté ou état commuté est un état stable défini, pour sa part, comme suivant immédiatement une transition " $y_i \rightarrow \overline{y_i}$ " de la fonction considérée. Les points de commutation correspondent, comme leur nom l'indique, aux états fugitifs de commutation qui apparaissent lors des transitions " $y_i \rightarrow \overline{y_i}$ ".

Il arrive cependant souvent qu'un point stable soit, à la fois, le point d'origine d'une transition " $y_i \rightarrow \overline{y_i}$ " et le point d'arrivée d'une transition " $\overline{y_i} \rightarrow y_i$ ". Il peut, dans ce cas, être aussi bien choisi comme point commutant que comme point commuté. Le choix se fait alors au mieux des simplifications possibles. Les graphes des figures 2.411 et 2.412 correspondent respectivement aux graphes d'émergence d'une seule fonction "mémoire" et d'une seule fonction "dibinaire".

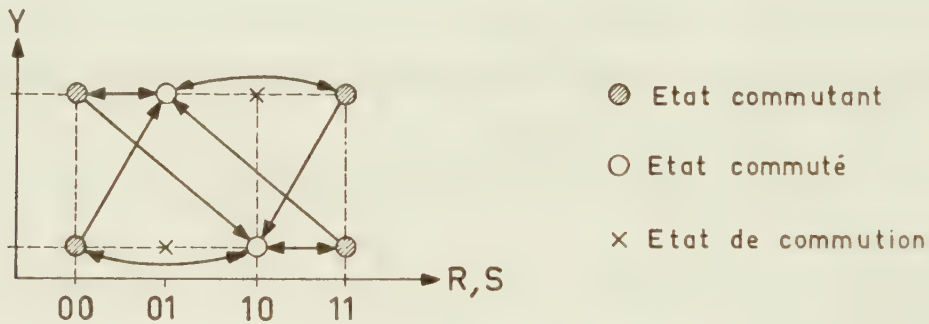


Figure 2.411 - Graphe de la fonction mémoire

$$y = \left| \begin{array}{c|c} y & \overline{R} \\ \hline \overline{R}, S & S \end{array} \right|$$

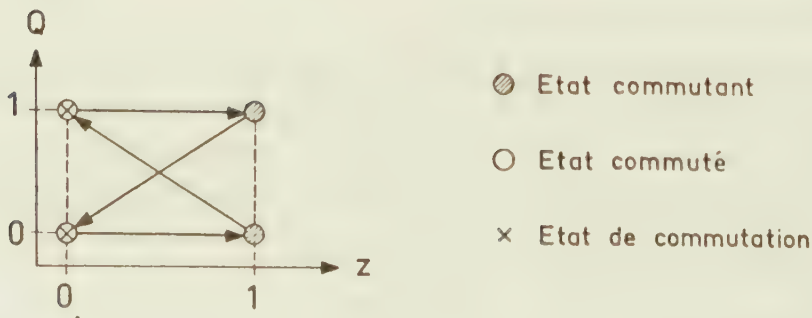


Figure 2.412 - Graphe de la fonction dibinaire inverse  $Q = Bi(Q, z)$ .

Rappelons que sur le graphe réflexe réduit, les fonctions "dibinaires" se distinguent des fonctions "mémoires" par le fait que leur graphe d'émergence comprend, au moins, un état stable superposé à un état de commutation comme dans le cas de la figure 2.412. D'une manière générale, la caractérisation des fonctions mémoires et dibinaires permet d'établir certaines règles qu'il est nécessaire d'observer pour procéder à une distribution convenable des points commutants et commutés d'un graphe d'émergence.

Ces règles ont été rigoureusement établies à l'aide de l'analyse binaire.

Lorsque les fonctions réflexes ont été calculées, on établit la fonction séquentielle par transcodage. Mais il reste cependant à déterminer les fonctions génératrices qu'il faut introduire dans les expressions, pour éliminer les aléas de commutation.

## 2.5 - Recherche et élimination des aléas de commutation

Il est possible de distinguer trois sortes d'aléas de commutation :

- les aléas de continuité
- les aléas de simultanéité
- les aléas de frontière.

Ces derniers, seuls, sont vraiment aléatoires et ne peuvent être éliminés. Mais leurs effets peuvent cependant, dans certains cas, être supprimés sans que la cause le soit.

Les aléas de continuité résultent du fait que les deux formes " $x$ " et " $\bar{x}$ " d'une même variable binaire, ne changent pratiquement jamais de valeur d'une façon simultanée. Ces aléas apparaissent donc, toutes les fois qu'une variable " $x$ " intervient dans une fonction binaire élémentaire " $y$ ", sous ses deux formes, directe et complémentée. La fonction  $y \in E_{01}$  est alors une fonction carrée biforme en " $x$ ".

Dans le cas où la technologie choisie correspond à un produit de produits

$$y = \left| \begin{array}{c} a.x \\ x.b \end{array} \right|, \text{ les aléas de continuité sont éliminés en mettant, dans la fonction, le consensus direct en facteur dual et en réalisant pratiquement } y = \left| \begin{array}{c} a.x \\ x.b \\ a.b \end{array} \right|$$

Si la technologie choisie mène par contre à un produit de produels,

$y = \left| \begin{array}{c} a \\ x \end{array} \right| \left| \begin{array}{c} x \\ b \end{array} \right|$ , c'est en mettant le consensus dual en facteur que l'on obtient l'élimination des aléas de continuité relatifs à " $x$ ". Il faut réaliser, dans le cas, la fonction " $y$ " sous la forme :

$$y = \left| \begin{array}{c} a \\ x \end{array} \right| \left| \begin{array}{c} x \\ b \end{array} \right| \left| \begin{array}{c} a \\ b \end{array} \right|$$

Les aléas de simultanéité sont dûs, en fait, à la non simultanéité des commutations de variables, ou de fonctions dépendant de mêmes variables, qui sont sensées, théoriquement, changer de valeur au même instant pour une transition particulière. La localisation, sur les expressions mathématiques binaires, des aléas de simultanéité, se fait en appliquant le théorème de non simultanéité suivant :

Les doubles transitions résultant de la non simultanéité des commutations de deux

éléments binaires, pour une fonction binaire, ne peuvent se produire que dans un produit ou un produel qui correspondent tous deux à la combinaison dans laquelle sont associées la valeur finale du premier élément qui commute et la valeur initiale de l'élément qui commute le dernier.

Considérons deux éléments binaires  $(y_1, y_2) \in E_{01}$ , et supposons, pour simplifier, que les commutations de " $y_2$ ", " $\bar{y}_2$ " se produisent toujours après celles de " $y_1$ ", " $\bar{y}_1$ ". Le théorème de non-simultanéité conduit alors aux implications suivantes :

$$\begin{aligned}
 [y_1 = (0 \rightarrow 1), y_2 = (0 \rightarrow 1)] &\Rightarrow [y_1 \cdot \bar{y}_2 = (0 \rightarrow 1 \rightarrow 0), \left| \begin{array}{c} \bar{y}_1 \\ y_2 \end{array} \right| = (1 \rightarrow 0 \rightarrow 1)] \\
 [y_1 = (0 \rightarrow 1), y_2 = (1 \rightarrow 0)] &\Rightarrow [y_1 \cdot y_2 = (0 \rightarrow 1 \rightarrow 0), \left| \begin{array}{c} \bar{y}_1 \\ \bar{y}_2 \end{array} \right| = (1 \rightarrow 0 \rightarrow 1)] \\
 [y_1 = (1 \rightarrow 0), y_2 = (0 \rightarrow 1)] &\Rightarrow [\bar{y}_1 \cdot \bar{y}_2 = (0 \rightarrow 1 \rightarrow 0), \left| \begin{array}{c} y_1 \\ y_2 \end{array} \right| = (1 \rightarrow 0 \rightarrow 1)] \\
 [y_1 = (1 \rightarrow 0), y_2 = (1 \rightarrow 0)] &\Rightarrow [\bar{y}_1 \cdot y_2 = (0 \rightarrow 1 \rightarrow 0), \left| \begin{array}{c} y_1 \\ \bar{y}_2 \end{array} \right| = (1 \rightarrow 0 \rightarrow 1)]
 \end{aligned}$$

S'il existe des états pour lesquels une fonction " $f$ " se trouve être égale à l'un des produits ou des produels qui figurent dans le second membre de l'une des implications précédentes alors que se produisent les transitions qui apparaissent dans le premier membre ; ces transitions engendrent alors dans " $f$ " des aléas de simultanéité qu'il est possible, en général, de supprimer en faisant intervenir simplement, en lieu et place de la fonction " $y_1$ ", la fonction délai  $\Delta_\tau(y_1)$  introduisant un retard " $\tau$ " suffisant pour que  $\Delta_\tau(y_1)$  commute après " $y_2$ " entraînant la disparition des doubles transitions  $(0 \rightarrow 1 \rightarrow 0)$  ou  $(1 \rightarrow 0 \rightarrow 1)$  de " $f$ ". Notons qu'une telle solution n'est possible que dans le cas où les éléments " $y_1$ " et " $y_2$ " sont interdépendants. Lorsque " $y_1$ " et " $y_2$ " sont indépendants, les aléas deviennent des aléas de frontière qui ne peuvent plus être éliminés bien qu'il soit possible, dans certains cas, d'en masquer les effets.

## 2.6 - Etude de l'échantillonneur d'impulsions aléatoires

La réalisation des opérateurs d'enchaînement a nécessité la résolution préalable de problèmes concernant les liaisons asynchrones. La fonction "échantillonneur" dont l'étude théorique va être développée, à titre d'exemple, dans ce paragraphe, sera utilisée pour la résolution de ces problèmes.

Le rôle du système consiste à prélever uniquement, parmi des trains d'impulsions aléatoires en largeur et en temps, la première impulsion dont la transition  $(0 \rightarrow 1)$  se présente après qu'un ordre " $a = 1$ " ait été donné.

Si la transition  $(0 \rightarrow 1)$  de " $a$ " se produit alors qu'une impulsion est en cours, c'est la suivante qui est considérée comme première impulsion et c'est elle qui doit en conséquence être prélevée. Toute impulsion prélevée doit l'être en totalité, même si des transitions de " $a$ " se produisent pendant le prélèvement. Les diagrammes de la figure (2.61) correspondent à

différents cas de fonctionnement.

Nous appellerons "x" la variable binaire aléatoire, "a" l'ordre de prélèvement et "S" le signal binaire de sortie qui correspond à l'impulsion prélevée,  $(a, x, S) \in E_{01}$ .

Nous constatons que le nombre total des états stables  $N_{st}$  est égale à 10. Pour que la condition des états soit satisfaite, il faut donc choisir deux fonctions réflexes élémentaires  $y_2 y_1$  ( $p = 2$ ). Nous avons alors,

$$N_{st} < 2^{n+p} \quad (10 < 2^4 = 16) \quad \text{avec}$$

$$N_{st} > 2^{n+p-1} \quad (10 > 2^3 = 8)$$

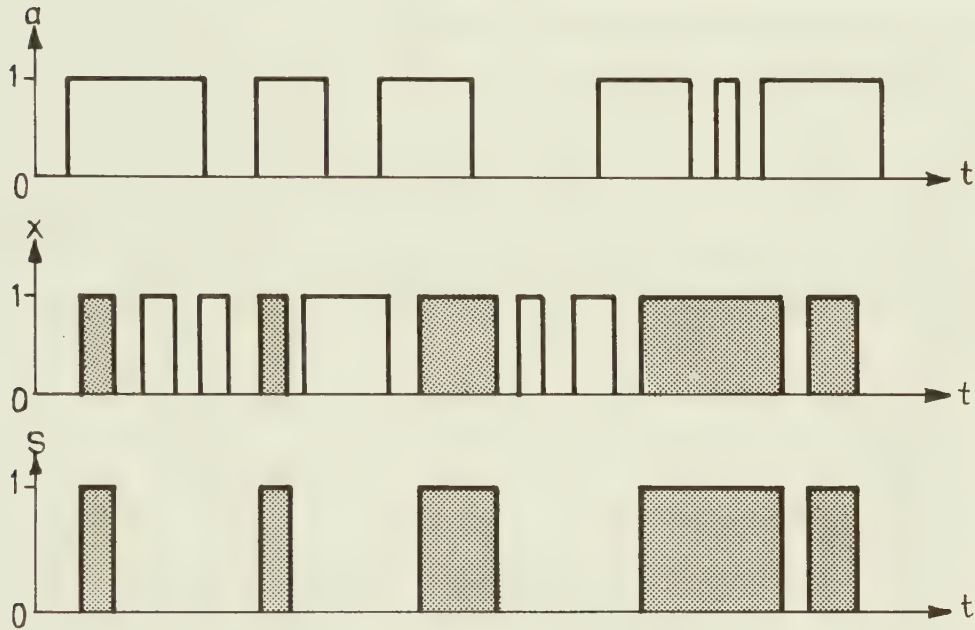


Fig. 2.61 - Diagramme des temps

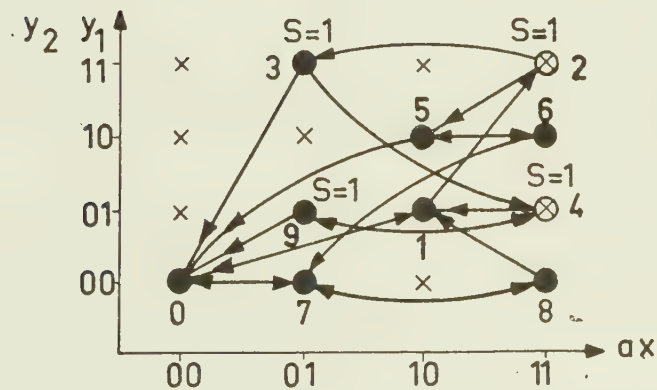


Fig. 2.62 - Graphe optimal

Pour la valeur  $ax=11$ , nous avons un maximum  $N_m$  d'états stables qui est égal à "4" (états 2, 6, 4, 8). Ce nombre maximal vérifie bien l'inégalité  $N_m \leq 2^P$  ( $4 = 2^2$ ).

Nous appellerons  $Y = y_2 y_1$ , le vecteur fonction du système et nous constatons, sur le graphe que deux états stables (2 et 4) sont superposés à des états de commutation. Cela signifie que la fonction  $y_2$  qui change de valeur entre ces deux états, est une fonction dibinaire.

En partant du graphe optimal, et en précisant les points commutés et les points commutants relatifs à  $y_2$  et  $y_1$  nous allons réaliser ces deux fonctions.

#### Réalisation de $y_1$ :

Il n'existe qu'une seule possibilité de choix qui consiste à prendre les points 0, 1 et 5 comme points commutés. Les autres points 2, 3, 4, 6, 7, 8 et 9 sont des points commutants.

Ce choix est noté sur le graphe de la figure 2.63.

L'état "1" fournit le groupe " $C_1$ "

$$C_1 \left( \begin{array}{l} a = 1, \quad x = 0, \quad y_2 = 0 \end{array} \right) \Rightarrow (y_1 = 1)$$

L'état "0" et l'état "5" constituent le groupe " $C_0$ ". Mais nous pouvons leur associer tous les états de commutation disponibles de la verticale " $ax = 00$ "; d'où

$$C_0 \left( \begin{array}{l} a = 0, \quad x = 0, \quad y_2 = 0 \\ a = 1, \quad x = 0, \quad y_2 = 1 \end{array} \right) \Rightarrow (y_1 = 0)$$

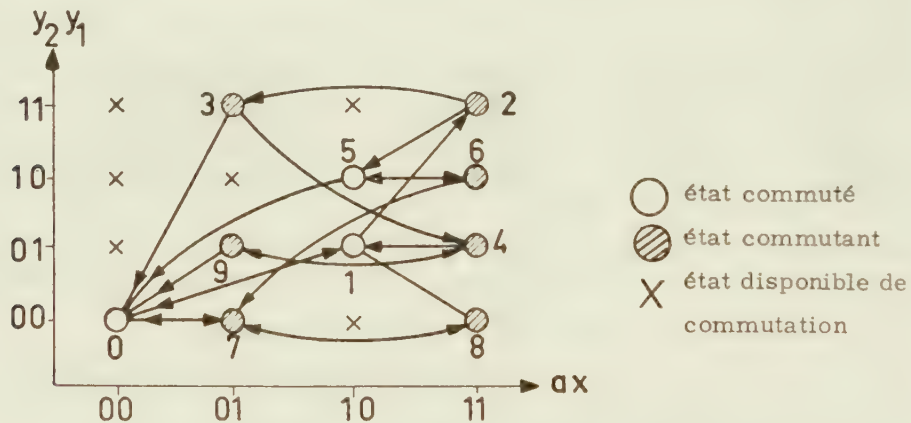


Fig. 2.63 - Graphe d'émergence relatif à  $y_1$

Les implications précédemment établies, permettent d'écrire immédiatement les fonctions d'entrée  $R_1, S_1$  de la fonction mémoire " $y_1$ "

$$R_1 = \bar{x} \left| \begin{array}{c} \bar{a} \\ y_2 \end{array} \right| \quad \text{et} \quad S_1 = a \cdot \bar{x} \cdot \bar{y}_2$$

$$y_1 = \left| \begin{array}{c} y_1 \\ S_1 \end{array} \right| \cdot \overline{R_1} = \left| \begin{array}{c} y_1 \cdot x \\ a \cdot \bar{x} \cdot \bar{y}_2 \end{array} \right|$$

Si nous désirons réaliser le circuit à l'aide de portes NAND par exemple, pour éviter les aléas de continuité, il faut introduire, en facteur dual dans cette expression, le

consensus direct "a.  $y_1$ .  $\bar{y}_2$ "

Nous obtenons en définitive :

$$y_1 = \begin{vmatrix} y_1 \cdot x \\ y_1 \cdot a \cdot \bar{y}_2 \\ a \cdot \bar{x} \cdot \bar{y}_2 \end{vmatrix}$$

Réalisation de  $y_2$  :

L'utilisation du graphe optimal revêt, toute son importance pour le calcul des éléments d'entrée  $R_2$ ,  $S_2$  et  $Z_2$  de la fonction dibinaire directe

$$y_2 = B(y_2, R_2, S_2, Z_2)$$

Notons qu'un parcours en forme de quadrilatère étoilé, comme le parcours 1-2-3-4, caractérise, à coup sûr une fonction dibinaire. Les points (2 et 4), qui correspondent à une superposition d'état stable et d'états de commutation, sont nécessairement des points commutés. Les états stables (1 et 3), qui représente les départs des commutations sont nécessairement des points commutants.

Sur le graphe de la figure 2.64, nous pouvons relever les combinaisons de valeurs qui correspondent aux trois points commutés 2, 4 et 9. Ces combinaisons constituent le groupe " $C_3$ ".

$$C_3 \left( \begin{array}{l} a = 1, \quad x = 1, \quad y_1 = 1, \quad y_2 = \emptyset \\ a = 0, \quad x = 1, \quad y_1 = 1, \quad y_2 = 0 \end{array} \right) \Rightarrow (y_2 \rightarrow \bar{y}_2)$$

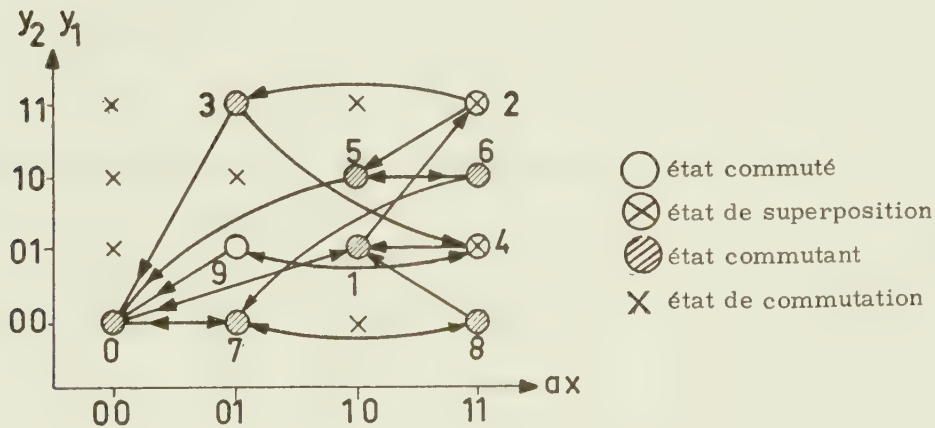


Fig. 2.64 - Distribution des états relatifs à la fonction dibinaire " $y_2$ "

Les passages de l'état "3" à l'état "0" de l'état "5" à l'état "0", de l'état "6" à l'état "7", sont des passages de point commutant à point commutant avec commutation de " $y_2$ " ( $y_2 = (1 \rightarrow 0)$ ). En associant, à l'état "7" et à l'état "0", tous les états de commutation de la verticale ax 00, nous pouvons constituer le groupe " $C_0$ ". :

$$C_0 \left( \begin{array}{l} a = 0, \quad x = 0, \quad y_1 = \emptyset, \quad y_2 = \emptyset \\ a = 0, \quad x = 1, \quad y_1 = 0, \quad y_2 = \emptyset \end{array} \right) \Rightarrow (y_2 = 0)$$

Il n'existe pas de groupe  $C_1$  et nous déduisons des implications précédentes, les expressions :

$$Z_2 = x \cdot y_1 \left| \begin{array}{c} a \\ \bar{y}_2 \end{array} \right| , \quad R_2 = \bar{a} \cdot \left| \begin{array}{c} \bar{x} \\ \bar{y}_1 \end{array} \right| , \quad S_2 = 0$$

Si nous utilisons une fonction dibinaire inverse  $y_2 = \text{Bi}(y_2, \bar{R}_2, \bar{S}_2, \bar{Z}_2)$  nous pouvons écrire finalement :

$$y_2 = \text{Bi}(y_2, \bar{R}_2, \bar{S}_2, \bar{Z}_2) ,$$

$$\bar{Z}_2 = \left| \begin{array}{c} \bar{x} \\ \bar{a} \cdot y_2 \\ \bar{y}_1 \end{array} \right| , \quad \bar{R}_2 = \left| \begin{array}{c} a \\ x \cdot y_1 \end{array} \right| , \quad \bar{S}_2 = 1 .$$

Il suffit maintenant, pour calculer la fonction séquentielle  $S$ , de rassembler les points pour lesquels  $S = 1$ , ce qui nous donne :

$$S = x \cdot y_1$$

Pour éliminer, en sortie, les aléas de commutation relatifs aux transitions théoriquement simultanées qui peuvent se produire, entre  $\bar{x}$ ,  $a$  et  $y_1$  d'une part, et  $x$ ,  $a$  et  $y_2$  d'autre part, il suffit d'introduire la fonction délai  $\Delta(x)$  dans les expressions de  $\bar{Z}_2$  et de " $S$ ". D'où les équations du circuit réalisé en pratique :

$$y_1 = \left| \begin{array}{c} y_1 \cdot \bar{x} \\ y_1 \cdot a \cdot \bar{y}_2 \\ a \cdot \bar{x} \cdot \bar{y}_2 \end{array} \right| ,$$

$$y_2 = \text{Bi}(y_2, \bar{R}_2, \bar{S}_2, \bar{Z}_2)$$

$$Z_2 = \left| \begin{array}{c} \Delta(x) \\ \bar{a} \cdot y_2 \\ \bar{y}_1 \end{array} \right|$$

$$\bar{R}_2 = \left| \begin{array}{c} a \\ x \cdot y_1 \end{array} \right|$$

$$\text{et } \bar{S}_2 = 1$$

$$S = y_1 \Delta(x)$$

Il est extrêmement intéressant de considérer le circuit obtenu comme une fonction de base que nous pourrions utiliser pour résoudre les problèmes fondamentaux de liaisons asynchrones telles que ceux posés par les opérateurs d'enchaînement.

Nous conviendrons donc, d'écrire symboliquement cette nouvelle fonction,

$$S = \text{Ech}(a, x)$$

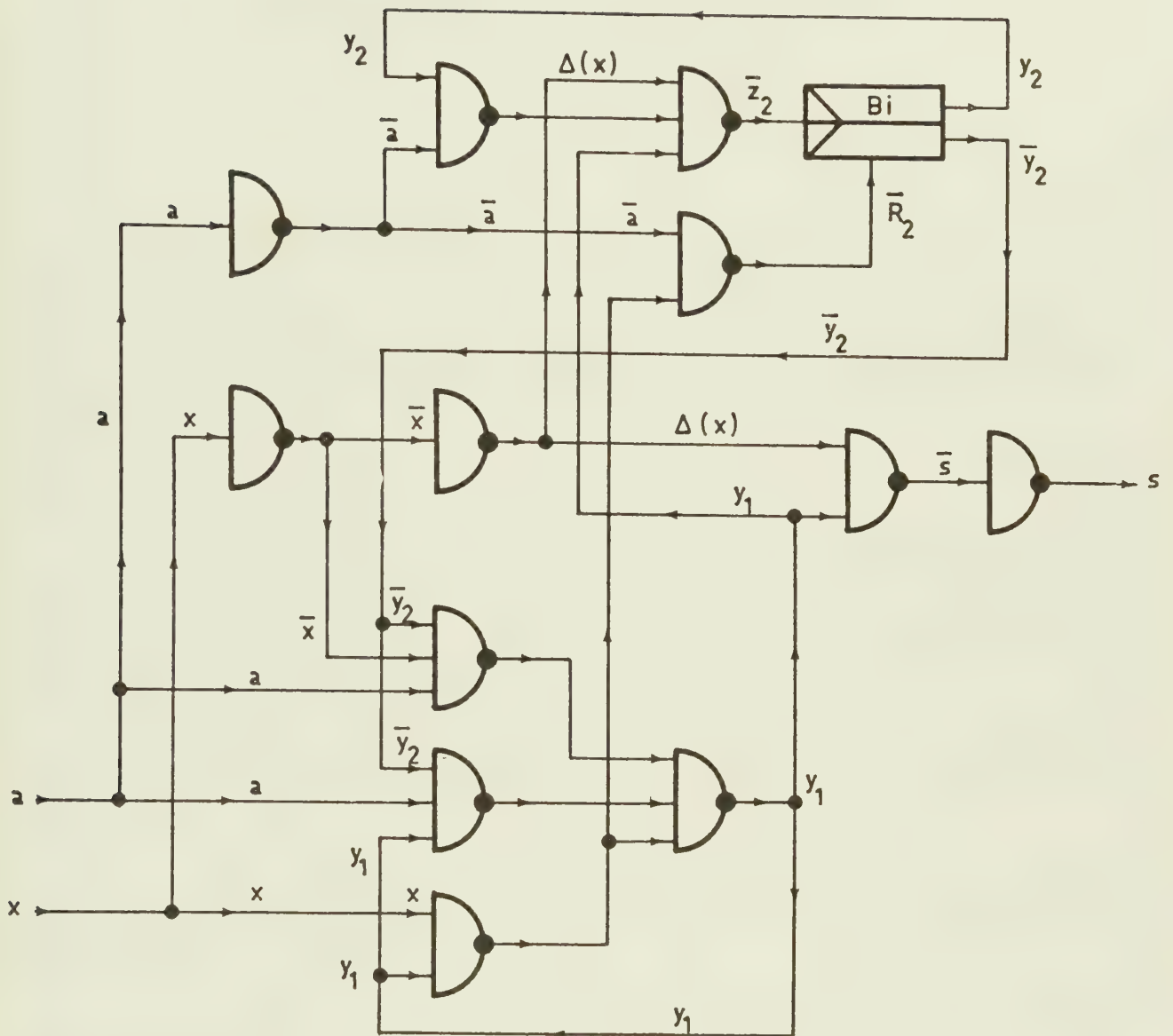


Fig. 2.65 - Schéma général de  $S = \text{Ech}(a, x)$  réalisé à l'aide d'une fonction binaire et de portes ON (NAND)



## CHAPITRE III

## ORGANISATION ACTUELLE DES CALCULATEURS ELECTRONIQUES

L'organisation actuelle des calculateurs comprend, dans ses grandes lignes, une structure dite "hardware" réalisée par les ingénieurs qui conçoivent le système et une structure dite "software" préparée par des programmeurs. La qualité et l'efficacité d'une telle organisation dépendent étroitement des relations qui existent entre ces deux sortes de structures fondamentales.

### 3.1 - Constitution du "software"

Le "software" qui réunit l'ensemble des algorithmes correspondant aux sous-programmes, programmes et macroprogrammes, a pour but de contrôler et de gérer le fonctionnement des blocs câblés ou "hardware", afin de résoudre les problèmes posés. Le "software" permet, en principe, d'utiliser toutes les possibilités offertes par la structure du calculateur avec une grande souplesse de fonctionnement.

Parmi les programmes "software" les plus courants, citons les assembleurs, les compilateurs, les sous-programmes I.O.C.S. (gestion entrées-sorties), les chargeurs et les superviseurs.

Il est possible, selon les besoins, d'imaginer un nombre pratiquement illimité de sous-programmes de service adaptés à différentes tâches usuelles telles que la diffusion, la reproduction, le tri, la duplication .....

La plupart des sous-programmes, y compris les I.O.C.S. et les superviseurs, sont, en fait, des opérateurs particuliers qui sont stockés dans la mémoire centrale sous la forme de programmes à tendance modulaire destinés à l'exécution, à des moments opportuns, de tâches généralement localisées.

Les sous-programmes I.O.C.S. sont utilisés pour toutes les opérations de transfert des données vers la mémoire centrale et pour la sortie des résultats vers les périphériques.

Le programme "superviseur" a, pour sa part, mission de gérer l'ensemble des activités de tous les sous-systèmes "hardware" du calculateur.

Les programmes "software" sont essentiellement évolutifs et peuvent être modifiés selon les besoins du traitement à la condition, toutefois impérative, de rester adaptés à la struc-

ture "hardware" du calculateur.

### 3.2 - Constitution du "hardware"

L'ensemble des systèmes câblés "hardware" d'un calculateur du type "batch processing" se compose essentiellement d'une mémoire centrale, d'une unité centrale, d'organes d'entrées-sorties et d'un bloc de commande centralisé.

- La mémoire centrale (M.C.), après l'assemblage, enregistre tous les programmes à exécuter ainsi que les résultats obtenus au cours du traitement.

Les blocs fonctionnels principaux associés à la mémoire centrale sont les suivants:

- le registre d'adresse "mémoire" (R.A.M.) qui enregistre les adresses de lecture ou d'écriture.

- le registre des données "mémoire" (R.D.M.) qui stocke les données à inscrire dans la mémoire ou les contenus à en extraire.

- l'unité de commande locale de la mémoire (U.C.M.) qui produit ou reçoit localement les signaux de commande.

- L'unité centrale (U.C.) est la pierre angulaire du système "batch processing" et comprend : l'unité arithmétique, les accumulateurs et certains registres spéciaux tels que les registres d'index.

- Les dispositifs périphériques sont, pour leur part, constitués par les enregistreurs et lecteurs magnétiques à bande, à disque ou à tambour, les perforateurs et lecteurs de cartes et de rubans perforés, les machines à écrire, les oscilloscopes de visualisation etc....

Des canaux de transmission assurent les liaisons entre les dispositifs "entrées-sorties" et la mémoire centrale. Chaque canal de transmission se compose d'un groupe de registres et de mémoires "tampons". Les données d'entrée ou de sortie sont normalement stockées dans les mémoires "tampons" (buffers) chargées d'effectuer le transfert entre les canaux et la mémoire centrale.

Les données sont groupées, distribuées, contrôlées par parité, converties par les interfaces puis transmises aux dispositifs récepteurs.

Plusieurs canaux sont généralement connectés à un seul équipement périphérique afin de donner plus de souplesse aux liaisons.

Dans le cas du "batch processing", les données sont rangées dans un ordre de priorité (stack-list) puis traitées à tour de rôle.

L'une des mémoires "tampons" reçoit les données, l'autre les transmet, assurant ainsi un traitement continu de l'information au niveau de l'unité centrale, indépendamment des opérations "d'entrées-sorties".

Tous les circuits d'un ordinateur sont sous la dépendance de l'unité de commande centrale qui contient quatre blocs fonctionnels :

- le compteur d'instruction (C.I.) qui affiche la commande à exécuter.

- le registre d'instruction (R.I.) qui enregistre les instructions de commande telles que la nature de l'opération, addition, soustraction, l'adresse de la première opérande, l'adresse de la deuxième opérande, l'adresse à laquelle les résultats doivent être transférés et

l'adresse de l'instruction suivante à exécuter.

L'analyseur est chargé de décoder la commande et sépare les caractères qui la compose. Il fournit les signaux qui ordonnent le déroulement du traitement. Le contrôleur examine les signaux de l'analyseur et distribue les tâches aux différents sous-systèmes dans un ordre convenable.

L'unité de commande centrale assure la gestion de toutes les communications et contrôle les activités de la mémoire centrale, de l'unité centrale et des dispositifs "entrées-sorties".

C'est l'unité de commande centrale qui prend les décisions d'attribution, lorsqu'un même sous-système "hardware" fait l'objet de demandes simultanées.

### 3.3 - Contrôle synchrone et asynchrone

Les concepts de contrôle centralisé ou de contrôle décentralisé sont intimement liés aux possibilités de réalisation d'un système sous forme synchrone ou sous forme asynchrone.

Dans le cas d'un système entièrement centralisé (contrôle synchrone) toutes les instructions sont fournies par un dispositif central de commande. Les opérations doivent être synchronisées à l'aide d'une horloge centrale unique. Les impulsions de l'horloge définissent les intervalles de temps, ou bits intervalles du système. Les différentes périodes de temps nécessaires à chaque opération sont toutes dérivées de l'horloge centrale, soit en introduisant des retards, soit à l'aide de compteurs qui accumulent un certain nombre de bits intervalles.

Pour pouvoir réaliser un système décentralisé il faut que les contrôles soient asynchrones. Un module de contrôle doit être associé localement à chaque bloc modulaire qui effectue le traitement de l'information proprement dite. Chaque opération dure le temps qui lui est nécessaire. Il suffit, lorsqu'elle est terminée, que le bloc modulaire transmette un signal de fin de traitement au module de commande qui, à son tour, peut ordonner le démarrage d'un nouveau traitement de l'information en attente. Cette information peut éventuellement provenir d'une pile ordonnée (stack-list).

#### 3.3.1 - Les raisons d'un contrôle asynchrone

Le contrôle, lorsqu'il est asynchrone, possède une autonomie qui lui confère une plus grande souplesse temporelle que le contrôle synchrone.

Dans le cas des systèmes asynchrones, toute opération, à l'intérieur d'un module, n'entraîne pas un temps d'occupation supérieur à celui nécessaire à son exécution. Elle n'affecte en rien le fonctionnement des autres éléments du système. Plusieurs modules d'une chaîne peuvent fonctionner en même temps. Certains peuvent être introduits ou retirés selon les nécessités du traitement. On peut imaginer un grand nombre de structures différentes utilisant les mêmes modules. Chacun d'entre eux peut recevoir un ordre d'interruption et être arrêté à un instant précis, ou quelquefois être mis hors circuit, sans affecter les autres modules voisins. C'est en cela que résident les qualités essentielles et l'intérêt des systèmes asynchrones qui ont conduit à la conception d'une organisation modulaire des ensembles de traitements de l'information.

Notons qu'il existe toujours, quelle que soit la conception des systèmes, certains

niveaux de fonctionnement où se posent inévitablement des problèmes asynchrones.

L'ignorance des caractéristiques précises des éléments de circuits, celle en particulier, de leurs temps de commutation et de propagation, l'impossibilité d'obtenir des signaux simultanés bien que synchronisés, conduit à l'existence d'intervalles de temps où les commutations se comportent de façon aléatoire ; d'où l'importance que revêt l'étude des aléas.

Si les problèmes asynchrones ne sont pas résolus, on démontre qu'en général la probabilité d'apparition d'un aléa dans le temps est proportionnelle au rapport entre le temps de montée des signaux mis en oeuvre et la période statistique moyenne d'apparition des conditions favorables à la production de cet aléa au cours du fonctionnement.

### 3.3.2 - Systèmes hybrides

Les problèmes asynchrones étant de toute façon posés, l'organisation d'un système peut être hybride au sens spatial autant que temporel.

C'est-à-dire qu'une partie de la machine peut être synchrone et l'autre asynchrone ou bien qu'une même partie peut fonctionner tantôt en synchrone et tantôt en asynchrone.

Dans le cas de la conception modulaire, les différents modules d'un système peuvent être pourvus d'une synchronisation interne autonome alors qu'il est impératif de conserver, entre eux, des liaisons de type asynchrone.

### 3.4 - Développement historique

Dans la première génération d'ordinateurs commercialisés, il n'était pas possible d'envisager le fonctionnement simultané des dispositifs "d'entrées-sorties" de la mémoire et de l'unité centrale.

Les programmes étaient exécutés les uns après les autres sans imbrication permise.

L'unité centrale était immobilisée pendant les opérations lentes des transferts "d'entrées-sorties" ; ce qui entraînait une perte de temps considérable.

C'est pourquoi les recherches ont été orientées, par la suite, vers l'optimisation du facteur d'utilisation de l'unité centrale et de la mémoire centrale qui sont les parties d'une machine dont le coût est le plus élevé.

Le modèle mathématique proposé par HOLLAND [11], visant à l'exécution simultanée de plusieurs programmes différents, a marqué le point de départ de projets du même genre.

La machine de HOLLAND disposait d'une unité de commande locale des dispositifs "d'entrées-sorties" mais il souffrait encore de nombreux défauts. Les idées développées par SOLOMON [9] conduisirent à un ordinateur dont la gestion centralisée, beaucoup trop rigide, ne permettait pas de résoudre convenablement tous les problèmes posés.

Des tentatives ont été faites, par la suite, pour associer les idées de HOLLAND et celles de SOLOMON, afin de réaliser une seule et même machine.

Les résultats ont été encourageants et le but partiellement atteint :

- en introduisant des mémoires "tampons" "d'entrées-sorties",
- en incorporant la multiprogrammation, réduite finalement au "time-sharing",
- en introduisant les possibilités de multi-traitement
- et en rassemblant, sous forme modulaire, les sous-programmes I.O.C.S. (entrées-sorties) et les sous-programmes du superviseur destinés à la gestion de l'ensemble des activités du système.

Le développement "d'ILLIAC IV" est un exemple significatif des tendances visant à la décentralisation de la gestion des systèmes.

"ILLIAC IV" est, en effet, un système à mémoire distribuée où une unité de mémoire est associée à chaque bloc de traitement. Le sous-système "d'entrées-sorties" exécute les opérations correspondantes, tandis que la gestion centrale est subdivisée en quatre unités de sous-contrôle dont chacune gouverne 64 blocs de traitement. Les unités de sous-contrôle peuvent fonctionner, soit simultanément, soit indépendamment les unes des autres ; ce qui permet aux quatre réseaux de blocs de traitement de fonctionner, soit isolément, soit groupés par paires, soit en concertation, selon la structure des données fournies. Les 64 blocs de traitement d'un réseau opèrent simultanément sous la gestion de leur unité de commande respective et sont ainsi du type synchrone relativement à cette dernière.

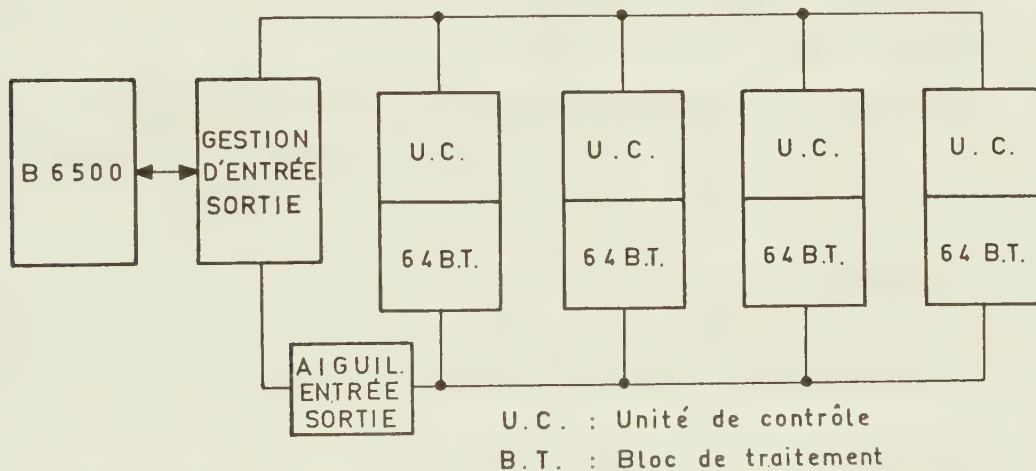


Fig. 3.41 : Synoptique simplifié du système " ILLIAC IV"

C'est donc un système qui peut être considéré comme semi-centralisé et par conséquent du type hybride.

Mais la situation ne semble pas en rester là car le développement du "L.S.I." qui permet de concentrer, à haute densité, un grand nombre de circuits, est prêt à mettre fin à la 3ème génération d'ordinateurs et permet d'envisager, très bientôt, une 4ème génération.

Dans ces ordinateurs d'avenir, la mémoire centrale est en voie de devenir une

mémoire active qui, avec l'unité de traitement, n'utilisera que quelques "chips" "L.S.I."

Les caractéristiques des machines de SOLOMON et HOLLAND sont, en fait, reprises pour obtenir une structure à unité de traitement distribuée qui a été conçue par Louis J. KOSZELA et Gray Y. WANG [10]. Cette unité de traitement se compose de cellules interconnectées formant un groupe. Les groupes sont ensuite interconnectés pour former le système entier. Une cellule n'est autre qu'un petit calculateur ayant une mémoire de faible capacité. Ce qui est intéressant, dans ce cas, c'est que le système est conçu pour une gestion à trois niveaux :

- un niveau de gestion élémentaire maintient la liaison entre cellules et contrôle leur fonctionnement individuel,
- un niveau de gestion de groupe contrôle toutes les cellules appartenant au même groupe,
- un niveau de gestion global contrôle enfin le fonctionnement de l'ensemble des groupes.

Chaque groupe peut donc fonctionner - soit indépendamment des autres groupes, soit avec eux, soit en concertation à la fois sous un contrôle de gestion local et un contrôle de gestion global.

R.L. VALLEE, dans un article publié au "B.I.N." n° 36 de Mars 1969, a posé les bases d'un type nouveau de gestion asynchrone. Le but proposé est également la décentralisation des systèmes de traitement numérique de l'information, mais en considérant une décomposition en blocs élémentaires qui peuvent être quelconques.

Chaque bloc exécute les opérations qui lui sont permises tandis qu'un opérateur, qui lui est associé, lui donne les ordres nécessaires.

Cet opérateur contrôle non seulement le fonctionnement du bloc mais maintient également la liaison avec les autres opérateurs, éventuellement en accord avec les instructions reçues d'un organe de gestion globale. Cet organe de gestion peut être prévu pour diriger l'ensemble des activités des blocs connectés selon les besoins de l'utilisation.

La solution proposée dans le "B.I.N." n° 36 était issue d'implications qui ne permettaient pas de résoudre le problème fondamental de liaison asynchrone. Les circuits obtenus ne pouvaient donc pas fonctionner correctement, sans envisager d'aménagements complémentaires.

C'est dans ce but que cette thèse m'a été confiée.

Les chapitres suivants sont consacrés à l'étude et à la réalisation d'un type particulier d'opérateur d'enchaînement dans lequel le problème asynchrone est résolu en utilisant le circuit d'échantillonnage aléatoire décrit au chapitre II. Ainsi, chaque bloc de traitement est rendu asynchrone et il se trouve capable de prendre en charge des données, disponibles à intervalles irréguliers, selon un rythme qui peut être totalement aléatoire.

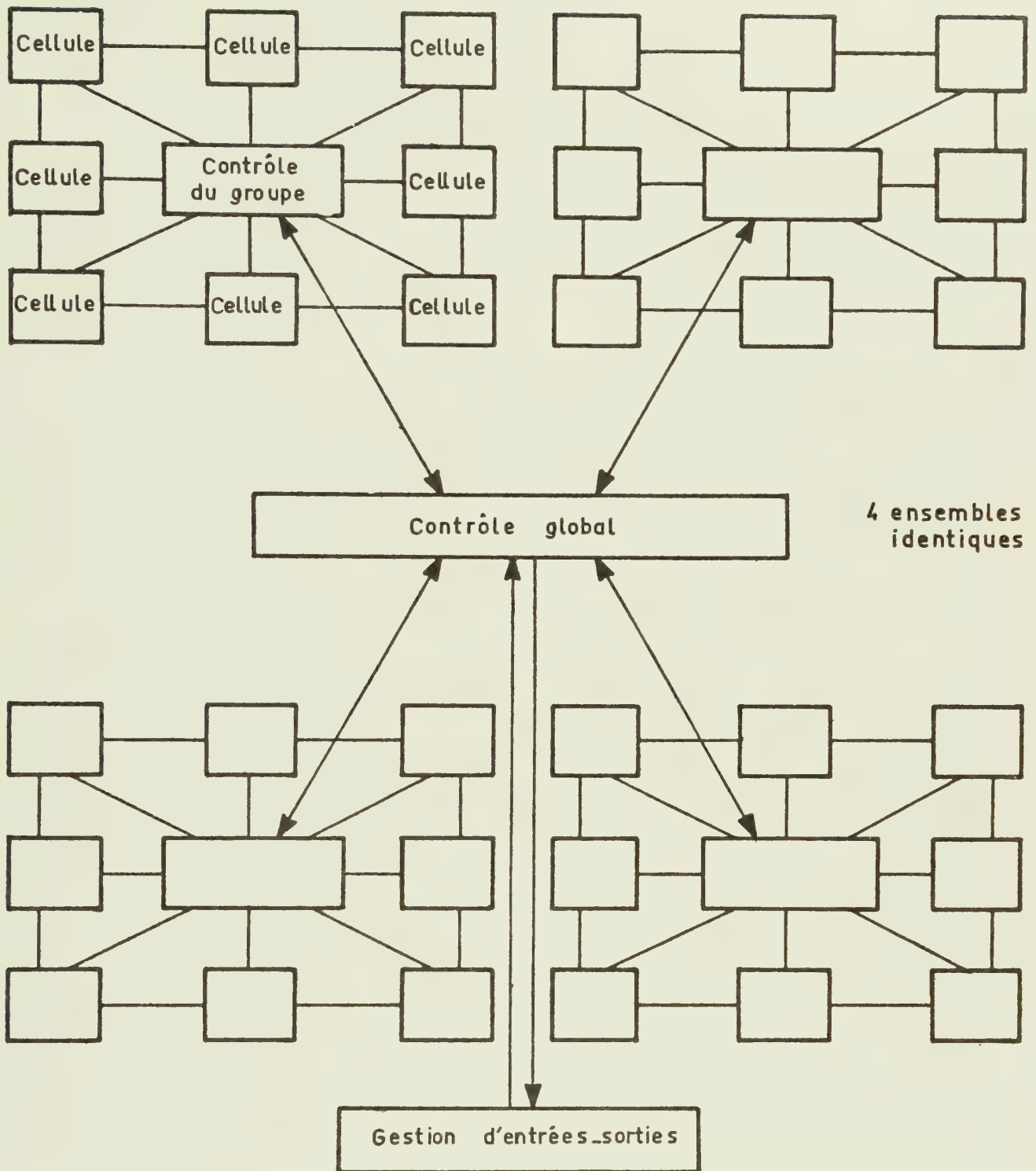


Fig. 3.42 - Schéma de principe du système de traitement distribué, utilisé par la NASA (Ref. 10)



## CHAPITRE IV

LES OPERATEURS D'ENCHAINEMENT  
FORME ELEMENTAIRE DES OPERATEURS PROGRAMMES4.1 - Généralités

La structure des systèmes automatiques est longtemps demeurée simple parce que la technologie ne permettait pas d'envisager la construction d'ensembles comprenant un grand nombre de composants. C'est pour cette raison que les créateurs de systèmes ont encore tendance à conserver à l'opérateur humain le rôle fondamental et déterminant qu'il a toujours tenu en priorité, ignorant trop souvent les possibilités de mémorisation, de décision et d'autonomie que possèdent les réalisations de la technique actuelle.

L'examen de ces possibilités conduit inévitablement à dissocier au niveau du traitement de l'information, la gestion et le traitement proprement dit.

Il est incontestable que l'évolution de la technique exige dès maintenant l'utilisation des moyens théoriques parfaitement adaptés à l'étude des automates numériques et particulièrement à cette gestion décentralisée indispensable au développement des systèmes de l'informatique. L'analyse binaire permet de remplir ce but et d'envisager la décentralisation en considérant les données acquises par un bloc particulier de traitement de l'information comme des résultats élaborés par un ou plusieurs blocs précédents.

Les résultats obtenus après traitement sont ensuite distribués à un ou plusieurs blocs suivants. La cascade réalisée peut être bouclée selon les nécessités du traitement. Si l'on désire en accord avec l'évolution des tendances de décentralisation, que chaque bloc puisse disposer d'une autonomie relative essentielle qui lui permette d'accomplir convenablement sa tâche relativement à la position qu'il occupe et en fonction des informations qu'il reçoit, il est alors nécessaire et impératif de lui associer en propre, un opérateur particulier, chargé d'assurer le contrôle, la surveillance de son fonctionnement global, et aussi de régir ses rapports avec les différents blocs émetteurs ou récepteurs qui peuvent lui être reliés au cours du fonctionnement.

Nous nous sommes fixé comme premier but de résoudre le problème dans le cas d'une simple cascade. Les opérateurs d'enchaînement, dans ce cas particulier, constituent une classe élémentaire des opérateurs programmes qui sont destinés à remplacer l'opérateur humain dans les systèmes automatiques évolués quel que soit le genre d'opérateur considéré : scientifique, industriel ou commercial.

La liaison en cascade se distingue des autres liaisons possibles par le fait que chaque bloc de traitement de l'information se trouve placé entre deux autres blocs ; l'un " $E_{n-1}$ " situé en amont, et l'autre  $E_{n+1}$ , situé en aval, qui prend en charge les informations traitées par le bloc  $E_n$ .

Nous supposons que le bloc " $E_n$ " dispose d'un registre d'entrée pour l'acquisition des données et d'un registre de sortie pour la mise à disposition des résultats, un seul registre pouvant éventuellement assumer les deux fonctions.

Il s'agit alors, en utilisant les moyens théoriques de l'analyse binaire, de définir et de calculer, puis de réaliser les circuits séquentiels qui vont constituer les opérateurs d'enchaînement cherchés.

Nous appellerons :

- C le signal de chargement du registre d'entrée,
- T le signal de commande du traitement de l'information,
- D le signal de mise à disposition des résultats,
- I le signal commandant l'interruption du fonctionnement,
- F le signal indiquant la fin du traitement,
- H un signal d'horloge éventuellement synchrone, local ou non, commandant les opérations de chargement.

Tous ces signaux sont de nature binaire  $(C, T, D, I, F, H) \in E_{01}$ .

Les indices associés à chaque symbole préciseront le bloc auquel il appartient

" $C_{n-1}$ " correspondra donc au signal de chargement du bloc " $E_{n-1}$ ", et " $C_n$ " au signal de chargement du bloc " $E_n$ ".

Nous conviendrons d'autre part d'utiliser les conventions suivantes :

- $T = 1$  engage le traitement et indique qu'il est en cours,
- $T = 0$  signifie qu'aucun traitement n'est en cours,
- $C = 1$  commande le chargement du registre d'entrée du bloc correspondant,
- $C = 0$  signale qu'aucun chargement n'est en cours,
- $D = 1$  indique que des résultats sont disponibles à la sortie du bloc correspondant,
- $D = 0$  aucun résultat n'est disponible,
- $I = 0$  interrompt le traitement en interdisant le chargement du registre d'entrée,
- $I = 1$  le bloc fonctionne normalement sans interruption.

Pour obtenir un fonctionnement correct du système d'interruption, il faut qu'au moment de la transition  $(1 \rightarrow 0)$  de " $I_n$ ", le bloc " $E_n$ ", s'il est en fonctionnement, termine le traitement en cours et suspende ensuite le fonctionnement ; cela est obtenu en ne chargeant plus le registre d'entrée aussi longtemps que " $I_n$ " reste égale à zéro. Le fonctionnement peut reprendre ensuite lorsque  $I_n = (0 \rightarrow 1)$ .

## 4.2 - Calcul d'un opérateur d'enchaînement

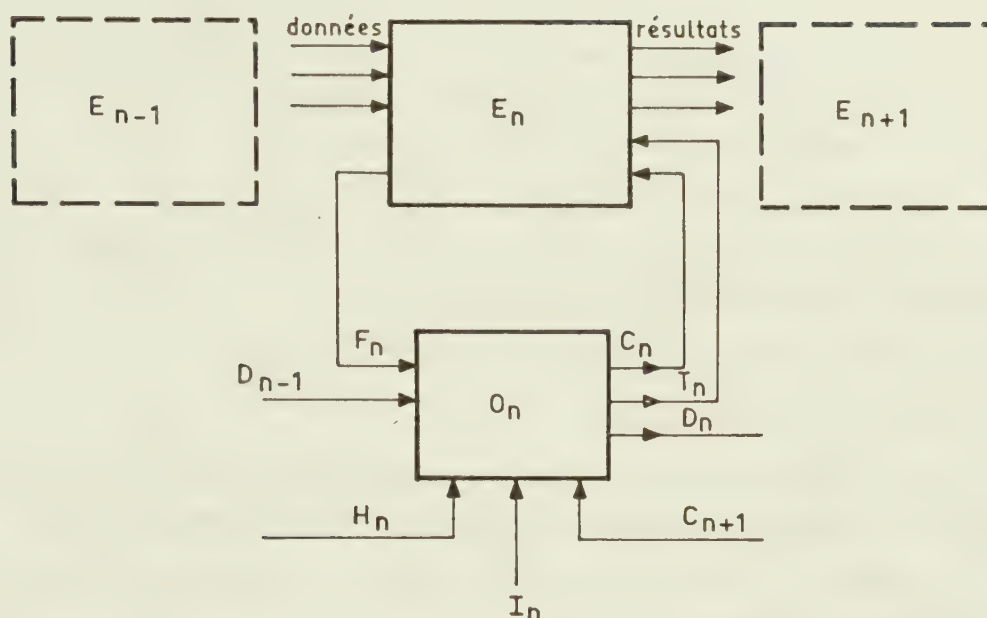


Fig. 4.21 - Diagramme montrant les paramètres, vecteurs variable et fonction, relatifs à l'opérateur d'enchaînement " $O_n$ ".

Définissons, en premier lieu, le vecteur variable de commande et le vecteur fonction de l'opérateur d'enchaînement " $O_n$ " associé au bloc " $E_n$ ".

Les composantes binaires du vecteur de commande sont au nombre de cinq. Il s'agit successivement de :

- $D_{n-1}$  information de mise à disposition des données sur le registre de sortie du bloc " $E_{n-1}$ " qui précède,
- $C_{n+1}$  signal de chargement du registre d'entrée du bloc " $E_{n+1}$ " qui suit,
- $I_n$  information de commande d'interruption,
- $F_n$  information de fin de traitement fournie par " $E_n$ ",
- $H_n$  signal d'horloge

L'opérateur d'enchaînement " $O_n$ " doit délivrer, d'autre part, le vecteur fonction qui comprend trois composantes binaires :

- $C_n$  l'ordre de chargement du registre d'entrée du bloc " $E_n$ ",
- $T_n$  l'ordre d'exécution des opérations de traitement que doit effectuer " $E_n$ ",
- $D_n$  l'information de mise à disposition des résultats sur le registre de sortie de " $E_n$ ".

Ainsi se trouvent définis, le vecteur de commande  $X_5 = D_{n-1}, C_{n+1}, I_n, F_n, H_n$  et le vecteur fonction séquentiel  $S_3 = C_n, T_n, D_n$ ,  $S_3 = \Psi(X_5)$ .

Nous allons maintenant calculer cette fonction séquentielle afin de réaliser, en pratique, l'opérateur d'enchaînement ainsi défini.

Considérons d'abord le cas particulier de la fonction binaire " $C_n$ ".

Pour donner l'ordre de chargement ( $C_n = 1$ ), il faut que soient assurées les conditions suivantes :

- $T_n = 0$  il n'existe pas de traitement en cours,
- $D_n = 0$  il n'existe pas de données à disposition à la sortie de " $E_n$ ",
- $D_{n-1} = 1$  des résultats sont disponibles à la sortie de " $E_{n-1}$ ",
- $I_n = 1$  aucun ordre d'interruption n'a été donné.

Comme il s'agit d'un cas de liaison asynchrone, nous allons envisager l'utilisation d'un échantillonneur d'impulsions aléatoires,  $C_n = \text{Ech}(A_n x_n)$ .

Il suffit de choisir, en fonction des conditions précédentes,

$$A_n = \overline{T_n} \cdot \overline{D_n} \cdot D_{n-1} \cdot I_n \quad \text{et} \quad x_n = H_n.$$

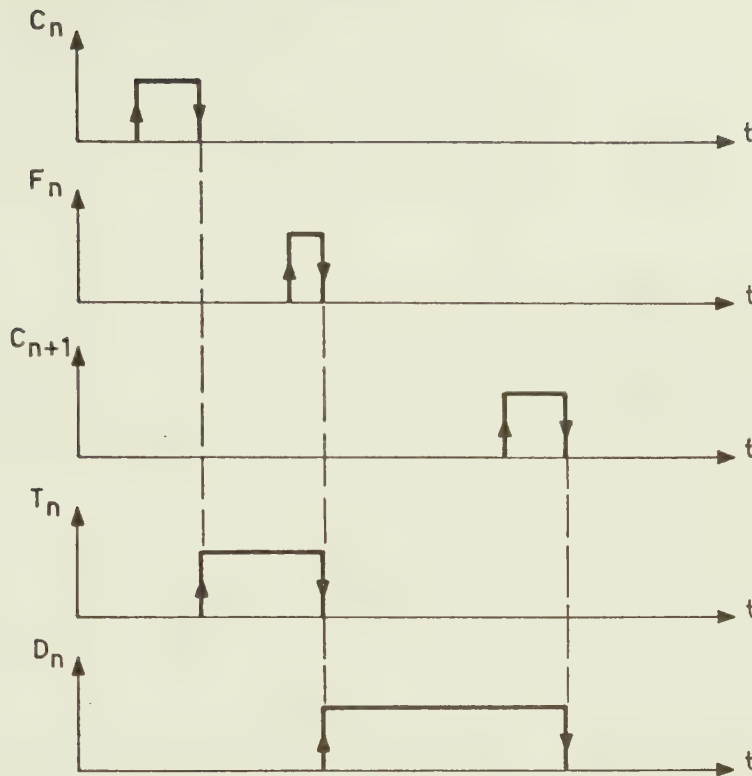
Le chargement du registre d'entrée s'effectuera correctement lorsque les conditions seront réunies, quelle que soit la façon dont elles se manifestent dans le temps.

Le signal " $C_n$ " durera toujours un temps égal à celui du signal horloge " $H_n$ " avec lequel il se trouve ainsi synchronisé.

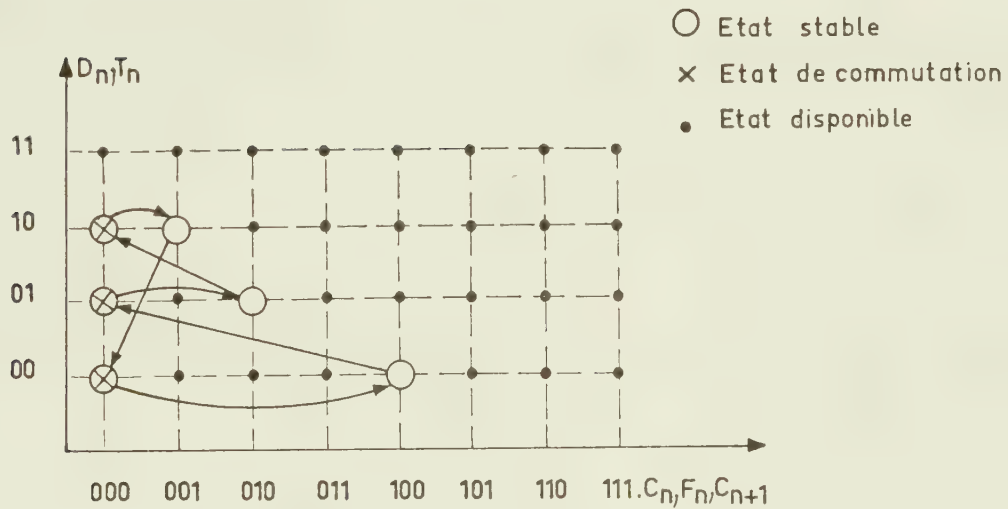
La fonction " $C_n$ " étant maintenant connue et le problème asynchrone résolu à l'aide du circuit échantillonneur, nous considérons " $C_n$ " comme une variable pour le calcul des fonctions binaires " $T_n$ " et " $D_n$ " qu'il reste à déterminer.

Pour ce faire nous pouvons établir un diagramme des temps représentant différentes phases de fonctionnement de " $T_n$ " et de " $D_n$ " ainsi que le graphe des états primitifs de la fonction séquentielle correspondante (figure 4.22).

" $O_n$ " ne doit donner l'ordre de traitement,  $T_n = (0 \rightarrow 1)$ , que dans le cas où le registre d'entrée de " $E_n$ " a été chargé,  $C_n = (1 \rightarrow 0)$ . " $T_n$ " doit ensuite être remis à zéro lorsque le signal de fin de traitement " $F_n$ " provenant de " $E_n$ " est transmis à l'opérateur. La fin du traitement indique que les résultats sont à disposition sur le registre de sortie et il faut alors que le signal de mise à disposition,  $D_n = (0 \rightarrow 1)$ , soit envoyé vers l'opérateur du bloc " $E_{n+1}$ " suivant, qui, lorsqu'il aura chargé son registre d'entrée,  $C_{n+1} = (1 \rightarrow 0)$  remettra " $D_n$ " à zéro. La fonction séquentielle " $D_n, T_n$ " se retrouve alors dans son état initial, prête à entamer une nouvelle séquence.



a - Diagramme des temps

Fig. 4.22 - Diagramme des temps et graphe des états primitifs de la fonction séquentielle  $(D_n, T_n) = \Sigma(C_n, F_n, C_{n+1})$ .

Le graphe des états primitifs fait apparaître six états stables et il n'existe jamais plus de deux états stables pour une valeur donnée de la fonction. Nous pouvons donc tracer un graphe réflexe optimal dans lequel la condition des états  $N_{st} \leq 2^{n+p}$  [3] est vérifiée, avec  $p = 2$  et  $n = 1$ , ( $6 < 2^3$ ).

Nous aurons alors une seule variable " $x_n$ " qui sera définie par le transcodage,  
 $x_n = \mathcal{C}(C_n, F_n, C_{n+1}, D_n, T_n)$ .

Le graphe réflexe optimal qui a été choisi est représenté fig. 4.23.

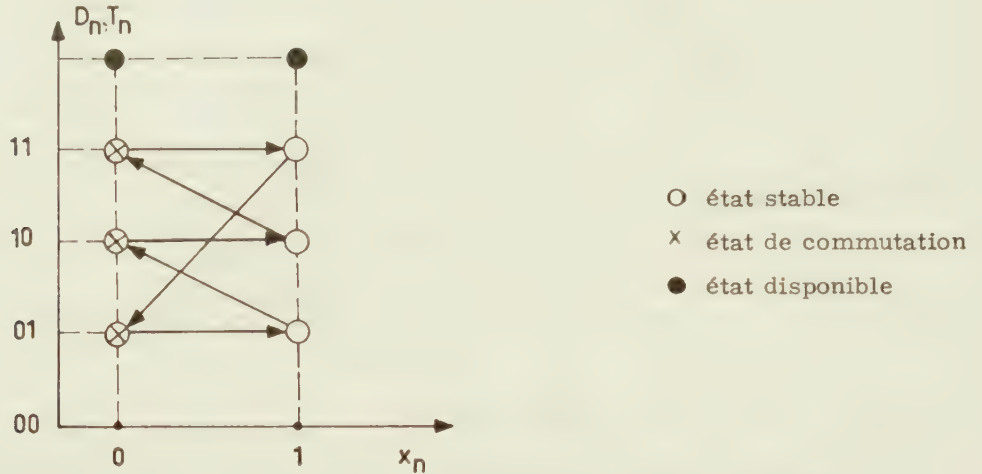
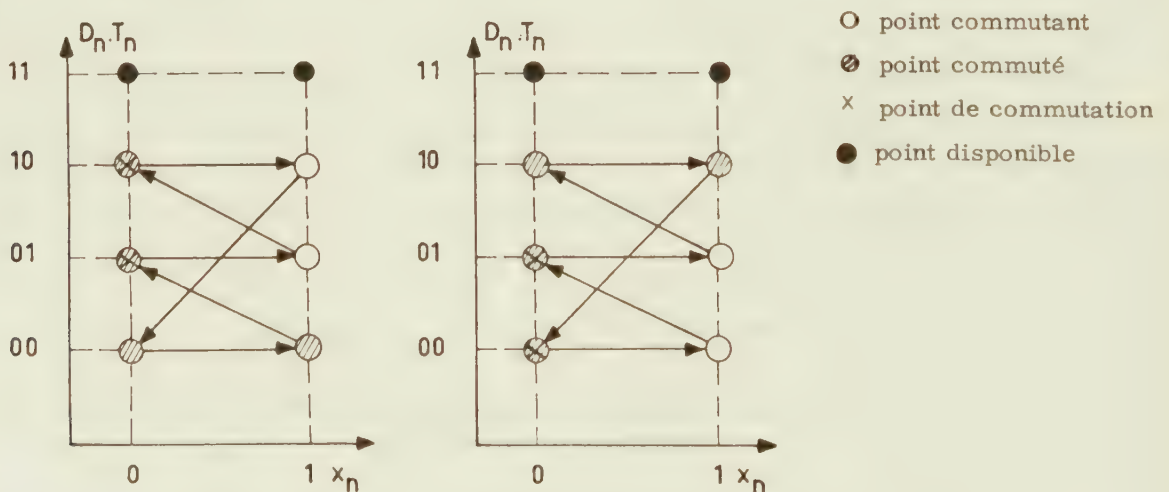


Fig. 4.23 - Graphe réflexe optimal de la fonction séquentielle " $D_n, T_n$ ".

Ce graphe est caractéristique d'une échelle de poids "3". Nous savons, en effet [3] que tout graphe séquentiel représentant un polygone fermé à sens de parcours unique peut être obtenu pratiquement à l'aide d'un transcodage d'échelle ; il est donc normal que le graphe réflexe optimal conduise à une échelle dans le cas étudié, puisque le graphe des états primitifs correspond justement à un polygone fermé à sens de parcours unique.

" $D_n$ " et " $T_n$ " sont des fonctions binaires caractérisées par la superposition d'états stables et d'états de commutation. Elles peuvent être calculées de différentes manières et en particulier par la méthode des graphes d'émergence [3] qui a l'avantage d'être exhaustive (figure 4.24).



a) - Graphe d'émergence de la fonction " $D_n$ "

b) - Graphe d'émergence de la fonction " $T_n$ "

Fig. 4.24 - Graphes d'émergences des fonctions " $D_n$ " et " $T_n$ ".

De ces graphes nous tirons immédiatement les deux fonctions dibinaires "D<sub>n</sub>" et "T<sub>n</sub>" en utilisant les coordonnées des points commutants associés à celles des points disponibles [3].

Nous obtenons :

$D_n = \text{Bi}(D_n, \bar{z}_0)$

$T_n = \text{Bi}(T_n, \bar{z}_1)$

$\bar{z}_0 = x_n$

$\bar{z}_1 = x_n \cdot \overline{D_n}$

$\left| \begin{array}{c} D_n \\ T_n \end{array} \right|$

Le même résultat peut être obtenu en dressant la table de vérité séquentielle réduite des fonctions dibinaires "D<sub>n</sub>" et "T<sub>n</sub>" et en repérant les états de préparation de basculement. Cette méthode permet le calcul des échelles synchrones en général [3].

|   | D <sub>n</sub> | T <sub>n</sub> | x <sub>n</sub> |                 |
|---|----------------|----------------|----------------|-----------------|
| 0 | 0              | 0+             | 1              |                 |
| 1 | 0+             | 1+             | 1              |                 |
| 2 | 1+             | 0              | 1              |                 |
|   | 1              | 1              | 1              | état disponible |

Il faut, pour terminer, calculer la fonction de transcodage,  
 $x_n = \mathcal{G}(C_n, F_n, C_{n+1}, D_n, T_n)$

Il suffit d'écrire une table de vérité réduite en utilisant le graphe des états primitifs. Cette table peut s'écrire :

| D <sub>n</sub> | T <sub>n</sub> | C <sub>n</sub> | F <sub>n</sub> | C <sub>n+1</sub> | x <sub>n</sub> |
|----------------|----------------|----------------|----------------|------------------|----------------|
| 0              | 0              | 1              | ∅              | ∅                | 1              |
| 0              | 1              | ∅              | 1              | ∅                | 1              |
| 1              | 0              | ∅              | ∅              | 1                | 1              |

Nous déduisons immédiatement, à partir de cette table de vérité, l'expression de x<sub>n</sub> :

$x_n =$

$\left| \begin{array}{c} \overline{D_n} \cdot \overline{T_n} \cdot C_n \\ \overline{D_n} \cdot T_n \cdot F_n \\ D_n \cdot \overline{T_n} \cdot C_{n+1} \end{array} \right|$

En remplaçant "x<sub>n</sub>" par sa valeur nous obtenons finalement :

$$\begin{aligned}
 D_n &= \text{Bi}(D_n, \bar{z}_0) & , & & \bar{z}_0 &= \left| \begin{array}{c} D_n \cdot \bar{T}_n \cdot C_{n+1} \\ \bar{D}_n \cdot T_n \cdot F_n \end{array} \right| \\
 T_n &= \text{Bi}(T_n, \bar{z}_1) & , & & \bar{z}_1 &= \left| \begin{array}{c} \bar{D}_n \cdot \bar{T}_n \cdot C_n \\ \bar{D}_n \cdot T_n \cdot F_n \end{array} \right|
 \end{aligned}$$

Si l'opérateur d'enchaînement est réalisé, comme c'est le cas, à l'aide de bascules correspondant à des fonctions dibinaires inverses et à l'aide de portes "ON" (NAND), il n'existe pas d'aléa de continuité dans les fonctions d'entrée " $\bar{z}_0$ " et " $\bar{z}_1$ ", le seul consensus non nul est relatif à " $\bar{z}_1$ "; il est égal à " $D_n \cdot C_n \cdot F_n$ " mais correspond à la fonction " $T_n$ " elle-même, et n'a pas lieu d'être utilisé dans ce cas.

Compte tenu de l'ordre dans lequel se produisent les commutations et de la technologie utilisée, l'application du théorème de non simultanéité [3] permet de déterminer les différents produits qui font apparaître des doubles transitions et risquent, par conséquent, de créer des perturbations dans le fonctionnement de l'opérateur calculé.

Dans l'ordre des séquences, les doubles transitions possibles sont les suivantes :

|                   |                                                               |
|-------------------|---------------------------------------------------------------|
| Commutations      | $C_n = (1 \rightarrow 0) , T_n = (0 \rightarrow 1)$           |
| Double transition | $\bar{C}_n \cdot \bar{T}_n = (0 \rightarrow 1 \rightarrow 0)$ |
| Commutations      | $F_n = (1 \rightarrow 0) , T_n = (1 \rightarrow 0)$           |
| Double transition | $\bar{F}_n \cdot T_n = (0 \rightarrow 1 \rightarrow 0)$       |
| Commutations      | $F_n = (1 \rightarrow 0) , D_n = (0 \rightarrow 1)$           |
| Double transition | $\bar{F}_n \cdot \bar{D}_n = (0 \rightarrow 1 \rightarrow 0)$ |
| Commutations      | $T_n = (1 \rightarrow 0) , D_n = (0 \rightarrow 1)$           |
| Double transition | $\bar{T}_n \cdot \bar{D}_n = (0 \rightarrow 1 \rightarrow 0)$ |
| Commutations      | $C_{n+1} = (1 \rightarrow 0) , D_n = (1 \rightarrow 0)$       |
| Double transition | $\bar{C}_{n+1} \cdot D_n = (0 \rightarrow 1 \rightarrow 0)$   |

Seul le produit " $\bar{T}_n \cdot \bar{D}_n$ " apparaît dans la commande de la fonction d'échantillonnage aléatoire qui fournit " $C_n$ ", et il faut, dans ce cas, éliminer l'aléa de simultanéité en retardant " $\bar{T}_n$ " dans l'expression,  $A_n = \bar{T}_n \cdot \bar{D}_n \cdot D_{n-1} \cdot I_n$ . Nous écrirons donc :

$$A_n = \Delta(T_n) \cdot \bar{D}_n \cdot D_{n-1} \cdot I_n.$$

Les équations générales de l'opérateur d'enchaînement s'écrivent finalement :

$$C_n = \text{Ech}(A_n, H_n)$$

$$A_n = \Delta(\overline{T}_n) \cdot \overline{D}_n \cdot D_{n-1} \cdot I_n$$

$$D_n = \text{Bi}(D_n, \overline{R}_0, \overline{S}_0, \overline{z}_0) \quad , \quad \overline{z}_0 = \left| \frac{D_n \cdot \overline{T}_n \cdot C_{n+1}}{\overline{D}_n \cdot T_n \cdot F_n} \right|$$

$$\overline{S}_0 = 1$$

$$T_n = \text{Bi}(T_n, \overline{R}_1, \overline{S}_1, \overline{z}_1) \quad , \quad \overline{z}_1 = \left| \frac{\overline{D}_n \cdot \overline{T}_n \cdot C_n}{\overline{D}_n \cdot T_n \cdot F_n} \right|$$

$$\overline{R}_1 = \overline{R}_0 \quad , \quad \overline{S}_1 = 1$$

La figure 4.25 représente le logigramme de chacune des fonctions binaires constituant l'opérateur d'enchaînement.

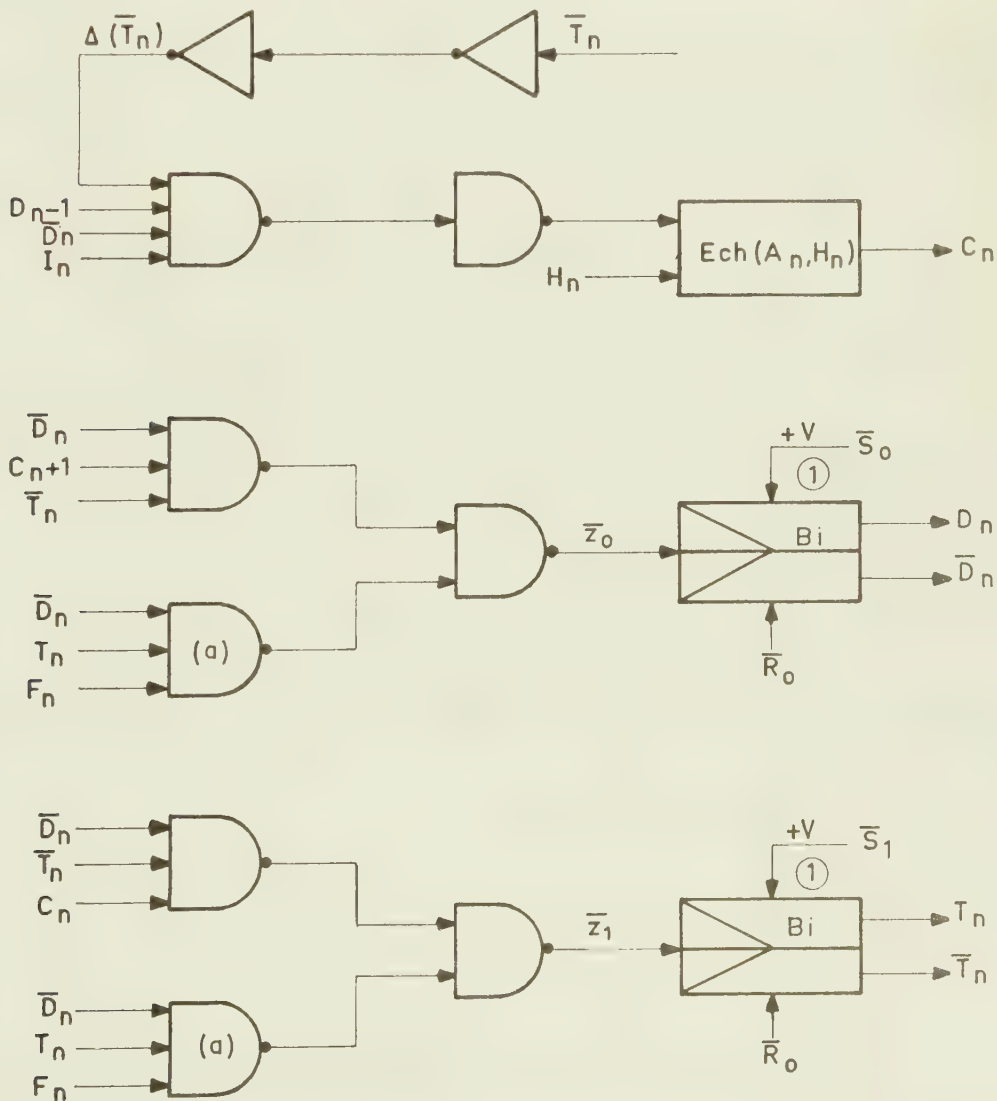


Fig. 4.25 - Logigramme des trois fonctions binaires qui constituent l'opérateur d'enchaînement.

Nous constatons qu'il existe un facteur dual commun aux deux fonctions d'entrée " $\overline{z}_0$ " et " $\overline{z}_1$ ". Ce facteur dual commun est le produit  $a = \overline{D}_n \cdot T_n \cdot F_n$ . Nous disposons, de plus, d'une variable supplémentaire " $R_0$ " qui permet une remise à zéro de l'opérateur.

Nous allons encore associer à l'opérateur d'enchaînement, deux fonctions importantes. L'une "sig" qui indique son bon fonctionnement et l'autre "dis" qui signale la disponibilité du bloc " $E_n$ ".

La fonction "sig" doit prendre la valeur "1" lorsque le fonctionnement s'effectue convenablement ; c'est-à-dire, lorsque deux fonctions, au moins, parmi  $C_n$ ,  $D_n$  et  $T_n$ , prennent la valeur zéro, d'où la table de vérité :

| $C_n$ | $D_n$ | $T_n$ | sig |
|-------|-------|-------|-----|
| 0     | 0     | 0     | 1   |
| 0     | 0     | 1     | 1   |
| 0     | 1     | 0     | 1   |
| 1     | 0     | 0     | 1   |

Nous en déduisons la fonction :

$$\text{sig} = \overline{C_n \cdot D_n} \cdot \overline{D_n \cdot T_n} \cdot \overline{T_n \cdot C_n}$$

Lorsque l'opérateur est disponible nous avons simultanément,  $C_n = 0$ ,  $D_n = 0$  et  $T_n = 0$ . Nous en déduisons la fonction "dis" égale à la valeur "1" lorsque ces conditions sont remplies :

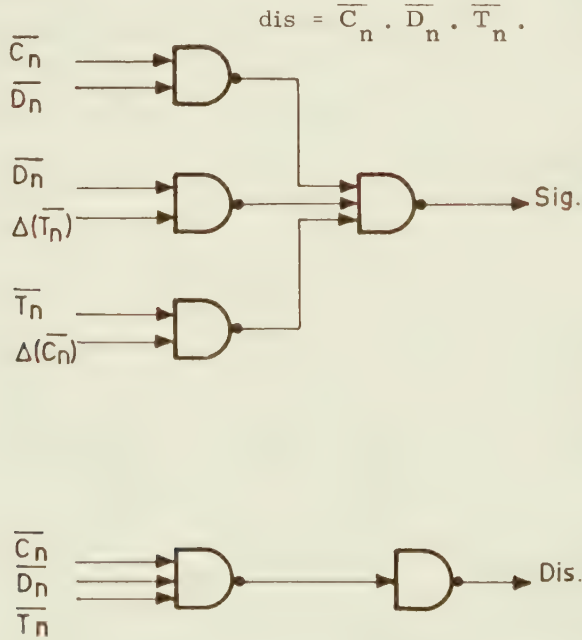


Fig. 4.26 - Logigrammes des fonctions de signalisation "sig" (bon fonctionnement) et "dis" (disponibilité des circuits).

Pour éviter les aléas de simultanéité, l'application du théorème de non simultanéité nous montre qu'il est nécessaire d'introduire des délais dans les produits  $\overline{D}_n \cdot \overline{T}_n$  et  $\overline{T}_n \cdot \overline{C}_n$  de manière à éviter les doubles transitions en écrivant :

$$\text{sig} = \begin{vmatrix} \overline{C}_n \cdot \overline{D}_n \\ \overline{D}_n \cdot \Delta(\overline{T}_n) \\ \overline{T}_n \cdot \Delta(\overline{C}_n) \end{vmatrix}$$

Lors du fonctionnement en cascade de plusieurs blocs, l'utilisation d'opérateurs d'enchaînement permet de traiter les informations avec une période pratiquement égale au temps d'occupation du bloc le plus lent, sous réserve que les informations se présentent sans interruption à l'entrée de la chaîne (figure 4.27).

Un programme séquentiel impose, par contre dans des conditions semblables, une période au moins égale à la somme des temps d'occupation de tous les blocs de la chaîne (figure 4.28).

Si  $t_1 < t_3 < t_2$ , les informations prises en charge à l'entrée peuvent se succéder avec des intervalles de temps très voisins du temps " $t_2$ " car les trois blocs peuvent fonctionner simultanément.

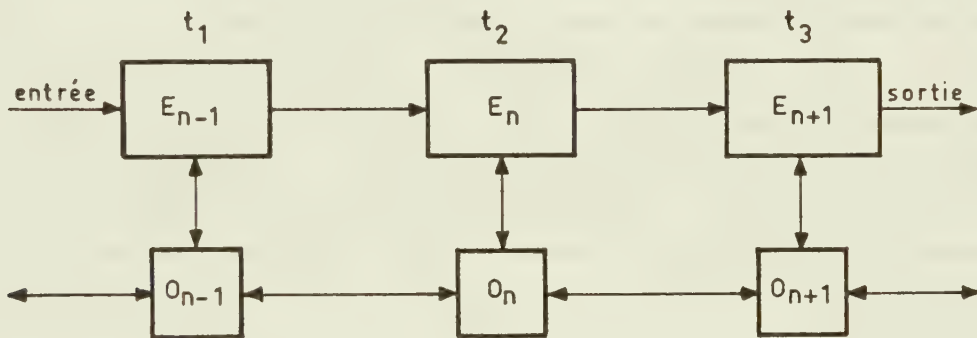


Fig. 4.27 - Blocs en cascade, contrôlés par des opérateurs d'enchaînement.

Dans le cas d'un programme séquentiel centralisé, les blocs fonctionnent successivement et les informations prises en charge se succèdent avec des intervalles de temps :

$$\Delta t \geq (t_1 + t_2 + t_3) .$$

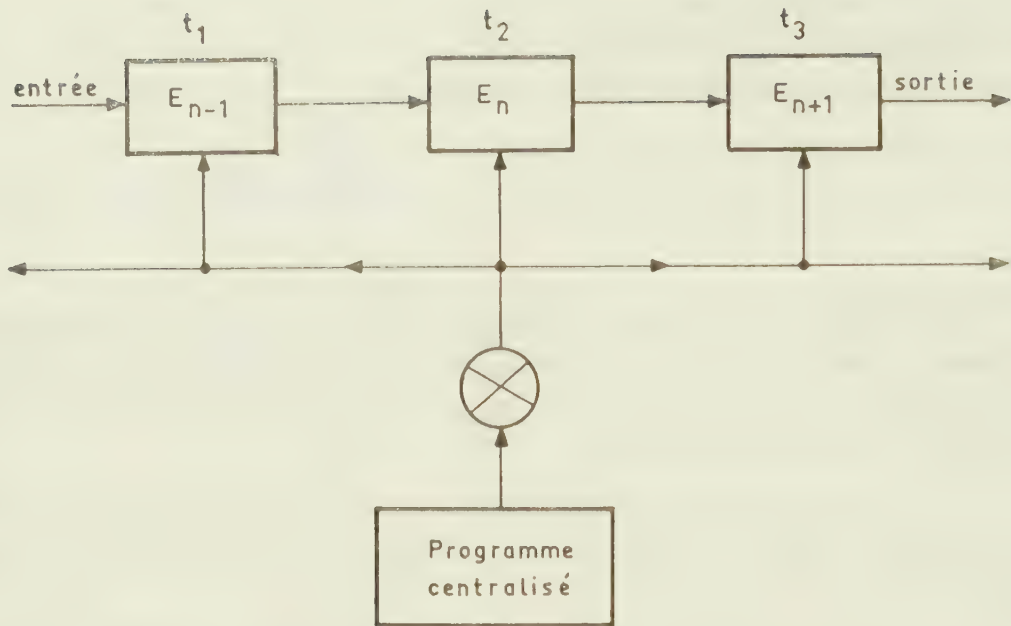


Fig. 4.28 - Blocs en cascade, contrôlés par un programme central.

Les opérateurs d'enchaînement calculés sont asynchrones ; il en résulte un temps de traitement qui dépend également de l'information traitée.

Avec les opérateurs d'enchaînement, il est possible d'envisager, dès maintenant, le développement d'opérateurs programmes permettant l'interconnexion de plusieurs ensembles de traitement d'information en adaptant leur fonctionnement aux différents modes d'opération désirés. L'étude et la réalisation de ces opérateurs programmes nous orientent vers une évolution nouvelle des systèmes de traitement de l'information, couvrant au-delà des calculateurs et des ordinateurs, le très vaste et important domaine des systèmes numériques.

#### 4.3 - Aspect statistique du fonctionnement des opérateurs d'enchaînement

Il est très intéressant de noter que la méthode P.E.R.T. est applicable aux opérateurs d'enchaînement et que cette méthode permet de calculer le temps de traitement moyen des différents blocs. Le procédé repose essentiellement sur trois estimations de temps :

- le temps optimiste "a"
- le temps le plus probable "m"
- le temps pessimiste "b"

Le temps de traitement moyen est alors défini par le rapport :

$$T_e = \frac{a + 4m + b}{6}$$

Ce rapport représente, en effet, le temps moyen que prendrait une opération lorsqu'elle se répète souvent.

La variance " $\sigma^2$ " est définie par la relation  $\sigma^2 = \left(\frac{b-a}{6}\right)^2$  ; relation qui est en fait la mesure de l'incertitude sur le temps que prendra, en général, une opération de période moyenne " $T_e$ ".

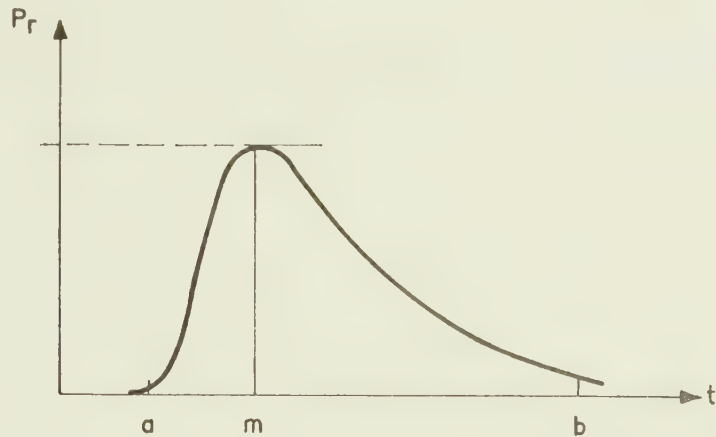


Fig. 4.31 - Courbe de probabilité définie à partir des trois temps a, m et b

Soit " $T_s$ " le temps théoriquement prévu pour l'ensemble des opérations d'un système, et " $T_E$ " la somme de tous les temps moyens relatifs aux différentes opérations du système.

Le facteur de probabilité " $Z$ " est donné par la relation,

$$Z = \frac{T_s - T_E}{\sqrt{\sum (\sigma_{T_E}^2)}} \quad , \quad \text{dans laquelle } \sum (\sigma_{T_E}^2) \text{ correspond à la somme des}$$

variances des temps d'opération servant au calcul de " $T_E$ " (en utilisant les chemins critiques seulement).

La probabilité " $P_r$ ", pour que le délai " $T_s$ ", théoriquement prévu, soit respecté dans une séquence complète, est donnée par la table de correspondance de  $Z$  et de  $P_r$  [4] et [5].



## CHAPITRE V

## REALISATION PRATIQUE DES OPERATEURS D'ENCHAINEMENT

5.1 - Composants utilisés

Les opérateurs d'enchaînement que nous avons réalisés en laboratoire utilisent des circuits intégrés T.T.L. (Transistor Transistor Logic), SN-7400N, SN-7410N, SN-7420N et SN-7472N, fabriqués par la Société "Texas-Instruments". Les circuits de la série T.T.L. comme ceux de la série D.T.L. (Diode Transistor Logic), utilisent fondamentalement des portes "ON" (NAND), mais la technologie T.T.L. a été choisie de préférence à la D.T.L., à cause d'une rapidité plus grande des circuits (30 MHz), de leur excellente stabilité et de leurs faibles variations de seuils en fonction de la température.

Ce choix a été également dicté par une bonne immunité au bruit, par un facteur pyramidal de sortie (sortance) élevé et un faible temps de propagation.

Ces circuits ont des impédances de sortie d'environ 70 ohms correspondant au "un" logique, et d'environ 10 ohms correspondant au "zéro" logique. Il s'ensuit un temps de réponse plus bref que celui des circuits D.T.L. qui, eux, ont une impédance de sortie de l'ordre de 600 ohms. La bascule J-K "maître-esclave", SN-7472N a été choisie, en particulier, parce qu'elle répond à la condition des seuils [3] qui caractérise les bascules associées, en pratique, aux fonctions dibinaires.

Rappelons que les caractéristiques principales des différents circuits de la série T.T.L. sont les suivantes :

|                                              |                           |
|----------------------------------------------|---------------------------|
| - tension d'alimentation                     | 4,5 à 5,5 volts           |
| - facteur pyramidal de sortie (fan-out)      | 10                        |
| - impédance de sortie (1 logique)            | 70 $\Omega$               |
| - impédance de sortie (0 logique)            | 10 $\Omega$               |
| - niveau de la tension de sortie (1 logique) | 2 à 4 volts               |
| - seuil minimal                              | 0,8 volts sur 50 $\Omega$ |
| - seuil de commutation nominal               | 1,5 volt                  |
| - impédance d'entrée (1 logique)             | 400 k $\Omega$            |
| - impédance d'entrée (0 logique)             | 4 k $\Omega$              |
| - puissance dissipée par porte               | 10 mW                     |

|                                               |        |
|-----------------------------------------------|--------|
| - puissance dissipée par bascule              | 40 mW  |
| - temps de propagation par porte              | 13 ns  |
| - temps de propagation par bascule            | 30 ns  |
| - temps de montée moyen (0 $\rightarrow$ 1)   | 2,5 ns |
| - temps de descente moyen (1 $\rightarrow$ 0) | 1,5 ns |

Le logigramme complet d'un opérateur d'enchaînement est donné par la figure 5.11 et le schéma de liaisons par la figure 5.12.

Trois opérateurs en cascade ont été connectés selon le schéma de la figure 5.13 et ont été montés sur un châssis standard "CAMAC" 1/25 [13]. Le système "CAMAC" est un système modulaire à tiroirs comprenant des plaquettes fonctionnelles qui peuvent être associées entre elles pour constituer un ensemble. Ces plaquettes fonctionnelles sont enfichables selon un standard électrique et mécanique conçu pour qu'il soit possible de connecter ou de déconnecter chaque plaquette suivant les nécessités du fonctionnement.

Les contacts des connecteurs sont disposés suivant une distribution standard qui groupe la transmission des données d'une part, les liaisons de contrôle et celles des alimentations d'autre part, indépendamment du type de tiroir utilisé.

Les photographies 5.14 et 5.15, relatives au tiroir construit pour les essais, représentent respectivement, une vue avant montrant aussi les circuits intégrés et une vue arrière montrant la partie câblée sur plaquette imprimée. Le tiroir contient trois opérateurs d'enchaînement. Chacun de ces opérateurs, réalisé sur une plaquette standard, dispose d'une signalisation de bon fonctionnement "BON" et d'une signalisation de disponibilité "DISP" ramenés, toutes deux, en face avant du tiroir, par l'intermédiaire de voyants de visualisation.

- Le voyant "disponibilité" s'allume lorsque le bloc contrôlé par l'opérateur correspondant est en mesure d'accepter de nouvelles données à traiter ; il s'éteint dès que le block procède à l'acquisition de ces nouvelles données, puis se rallume aussitôt que les résultats ont été pris en charge par le bloc suivant en attendant une nouvelle acquisition..

- Le voyant "bon fonctionnement" reste allumé tant que les opérations de traitement et, en général, le fonctionnement de l'opérateur restent corrects ; il s'éteint dès qu'une anomalie se présente dans le fonctionnement.

Les circuits de signalisation de "bon fonctionnement" et de "disponibilité" permettent ainsi de détecter, au niveau de chaque opérateur, certaines pannes de fonctionnement, lorsqu'elles se produisent.

A noter que dans le cas d'un ordre d'interruption transmis à l'opérateur, les deux voyants restent allumés aussi longtemps que se trouve maintenu cet ordre d'interruption. Précisons, toutefois, que la signalisation prévue ne concerne que le fonctionnement des opérateurs et qu'il faut lui adjoindre dans un système donné, la signalisation relative au traitement proprement dit de l'information, au niveau du block contrôlé.

## 5.2 - Principales mesures effectuées sur les opérateurs d'enchaînement réalisés

Les mesures et les contrôles ont été faits à l'aide d'un oscilloscope (Tektronix 585-A) équipé d'un tiroir à quatre voies (type M).

Les caractéristiques suivantes ont été observées en utilisant une tension d'entrée nominale de 5 volts pour le "un" logique sous une impédance de 50 ohms et un niveau minimal de 0,8 volt.

### - Caractéristiques des impulsions d'entrée :

|                        |       |
|------------------------|-------|
| - largeur à mi-hauteur | 80 ns |
| - temps de montée      | 20 ns |
| - temps de descente    | 20 ns |

### - Caractéristiques observées en sortie sur C, T et D :

|                     |       |
|---------------------|-------|
| - temps de montée   | 28 ns |
| - temps de descente | 40 ns |

Les caractéristiques réelles des impulsions observées doivent, en fait, être un peu meilleures, car il y a lieu de tenir compte de la bande passante propre au tiroir amplificateur de l'oscilloscope (30 MHz) qui correspond à un temps de montée d'environ 12 ns ayant le même ordre de grandeur que celui des circuits étudiés.

## 5.3 - Montage expérimental de simulation

Chaque bloc de traitement peut être simulé par un circuit comprenant d'une part un élément mémoire "M" qui enregistre l'ordre de chargement "C", une fonction délai "Z" dépendant de "T" et dont le retard " $\tau$ " simule le temps de traitement, une fonction impulsion de "Z" qui délivre, d'autre part à l'opérateur d'enchaînement correspondant, le signal de fin de traitement "F".

Pour calculer ces fonctions, il suffit de mettre les conditions énoncées, sous la forme d'implications [3].

### - Fonction mémoire "M"

$$\begin{aligned} [C = 1, Z = \emptyset] &\Rightarrow M = 1 \\ [C = 0, Z = 1] &\Rightarrow M = 0 \end{aligned}$$

### - Fonction délai $\Delta Z$

$$[M = 1, T = 1] \Rightarrow Z = 1_{(\tau)}$$

### - Fonction impulsion $I_{\tau_1}(Z)$

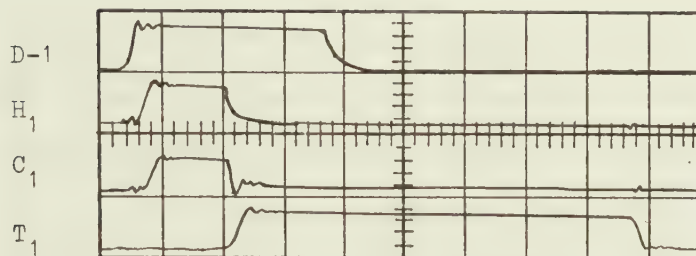
$$Z = (0 \rightarrow 1) \Rightarrow F = (0 \rightarrow 1, 1 \rightarrow 0) \tau_1$$

La fonction "M" est une fonction mémoire dans laquelle [3]  $R = \bar{C} \cdot Z$ ,  $S = C$ .

Les équations très simples du bloc simulé s'écrivent finalement :

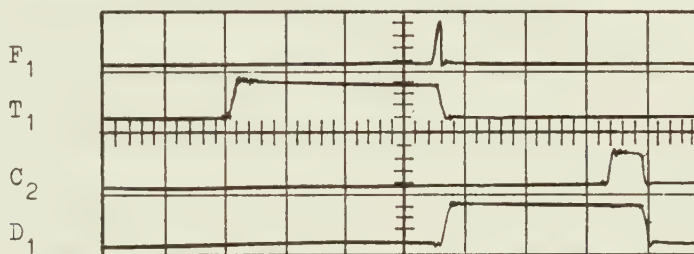
$$M = \begin{vmatrix} M \\ C \end{vmatrix} = \begin{vmatrix} \bar{Z} \\ C \end{vmatrix} = \begin{vmatrix} M \cdot \bar{Z} \\ C \end{vmatrix}$$



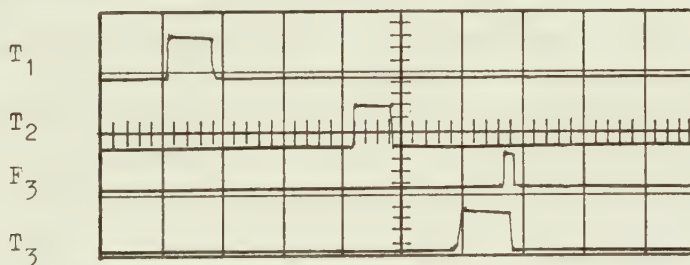


D-1 : Signal indiquant la disponibilité des résultats à la sortie de l'ensemble qui précède le premier bloc.

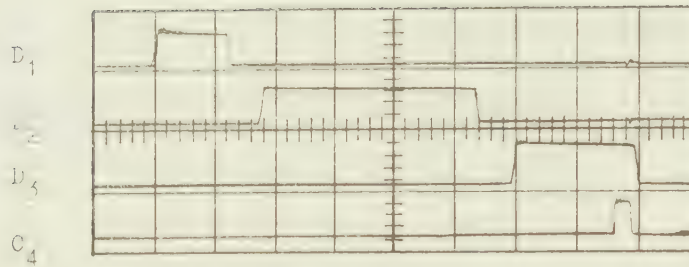
- Photo n° 1 Le premier bloc "1" est prêt à prendre en charge les données. L'impulsion de l'horloge "H<sub>1</sub>" charge le registre d'entrée et déclenche ensuite, en fin de charge, la séquence de traitement de l'information "T<sub>1</sub>".



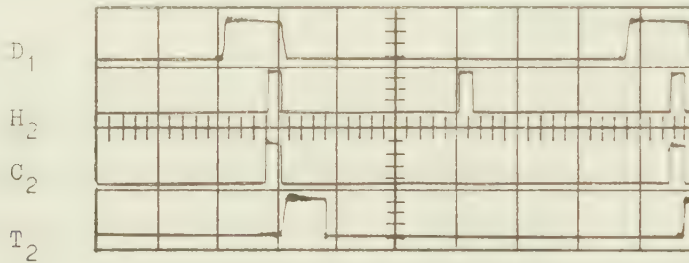
- Photo n° 2 Le signal de fin de traitement "F<sub>1</sub>" remet à zéro l'ordre "T<sub>1</sub>" et commande la transition (0 → 1) du signal de mise à disposition "D<sub>1</sub>" du bloc "1" correspondant. L'ordre de chargement "C<sub>2</sub>" du registre d'entrée du bloc "2" remet à zéro l'information "D<sub>1</sub>".



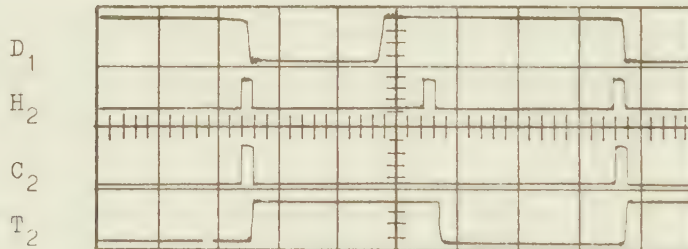
- Photo n° 3 Les trois ordres de traitement de l'information "T<sub>1</sub>", "T<sub>2</sub>", et "T<sub>3</sub>" sont déclenchés en cascade. Le signal "F<sub>3</sub>" remet à zéro "T<sub>3</sub>".



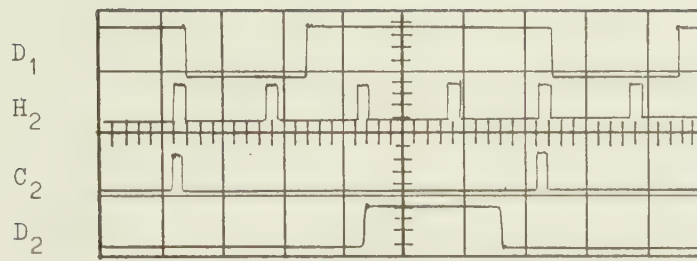
- Photo n° 4 Dans ce cas, sont représentés les trois signaux successifs de mise à disposition des données "D<sub>1</sub>", "D<sub>2</sub>" et "D<sub>3</sub>", ainsi que la remise à zéro de "D<sub>3</sub>" par la transition (1 → 0) de "C<sub>4</sub>" qui représente le signal suivant de prise en charge des résultats.



- Photo n° 5 Cette photo montre que le chargement d'un registre d'entrée ne peut s'effectuer que dans le cas où les résultats fournis par le bloc précédent sont mis à disposition (D = 1).



- Photo n° 6 Le second bloc ne peut prendre en charge les données précédentes lorsque le traitement est en cours. On peut remarquer, dans ce cas, que le premier et le second fonctionnent simultanément pendant une partie de la séquence étudiée.



- Photo n° 7 Le second bloc ne peut prendre en charge les données du précédent que dans le cas où le suivant a lui-même pris en charge les résultats qui sont mis à sa disposition.

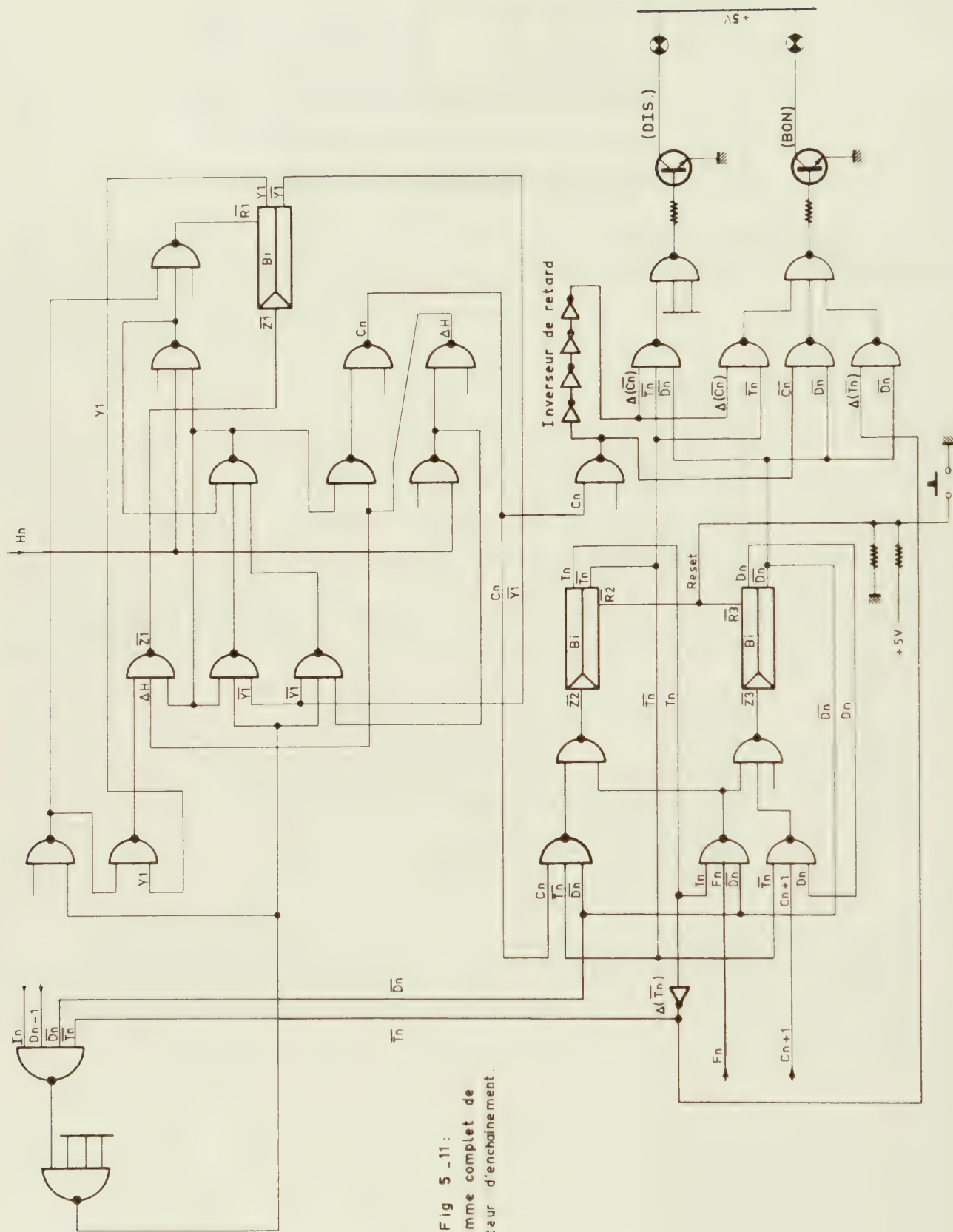
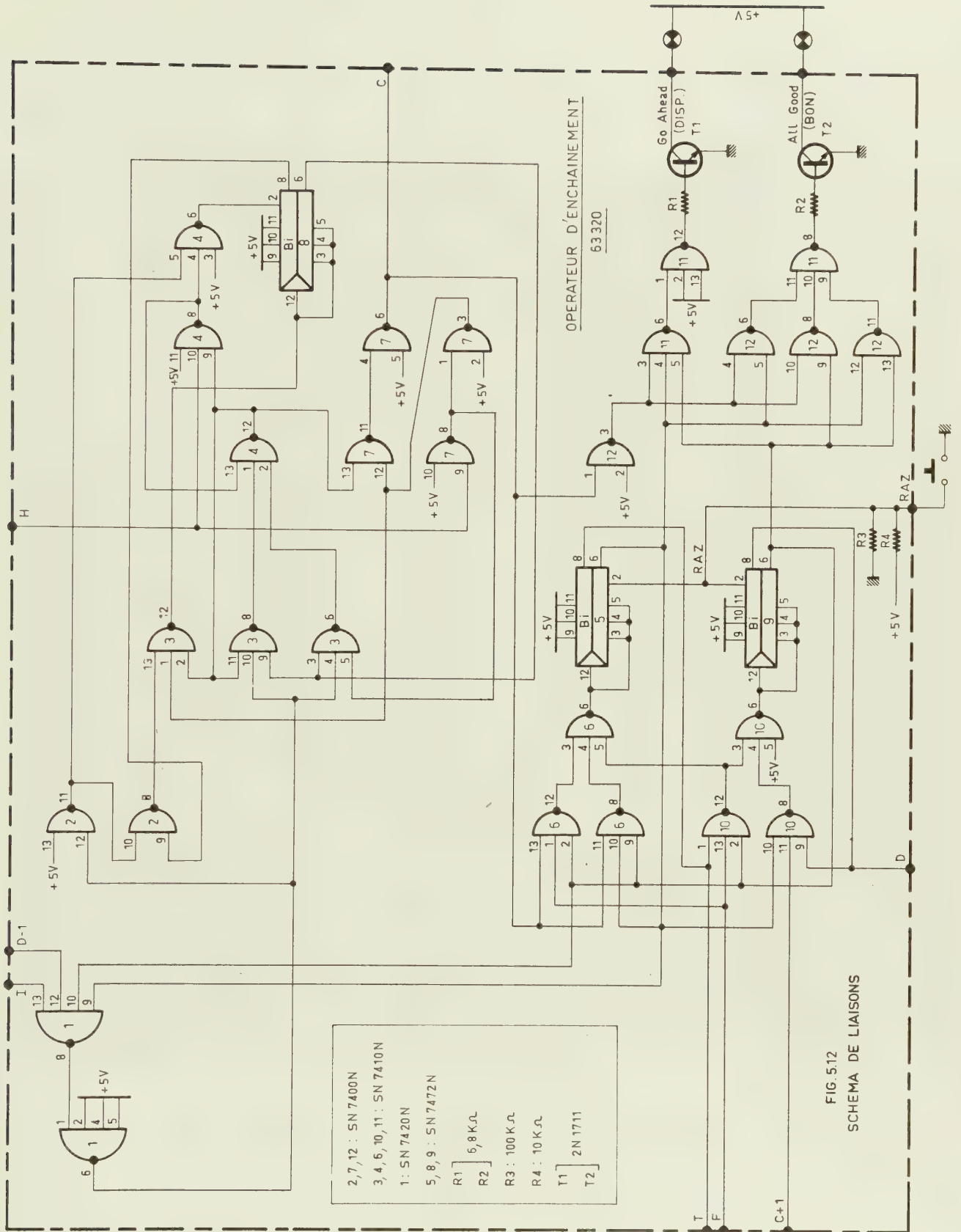


Fig 5 \_11 :  
Logigramme complet de  
l'opérateur d'enchaînement.



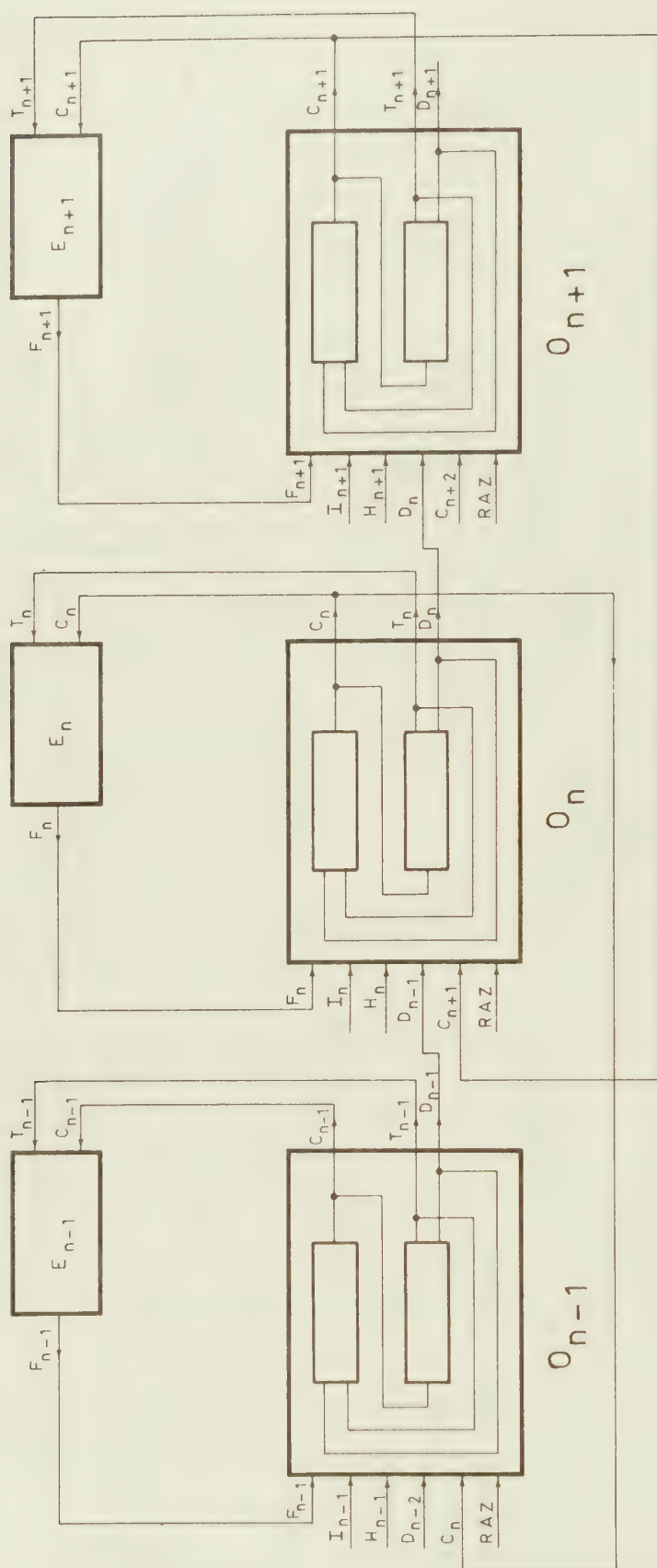


Fig 5.13 : Schéma de 3 opérateurs en cascade.

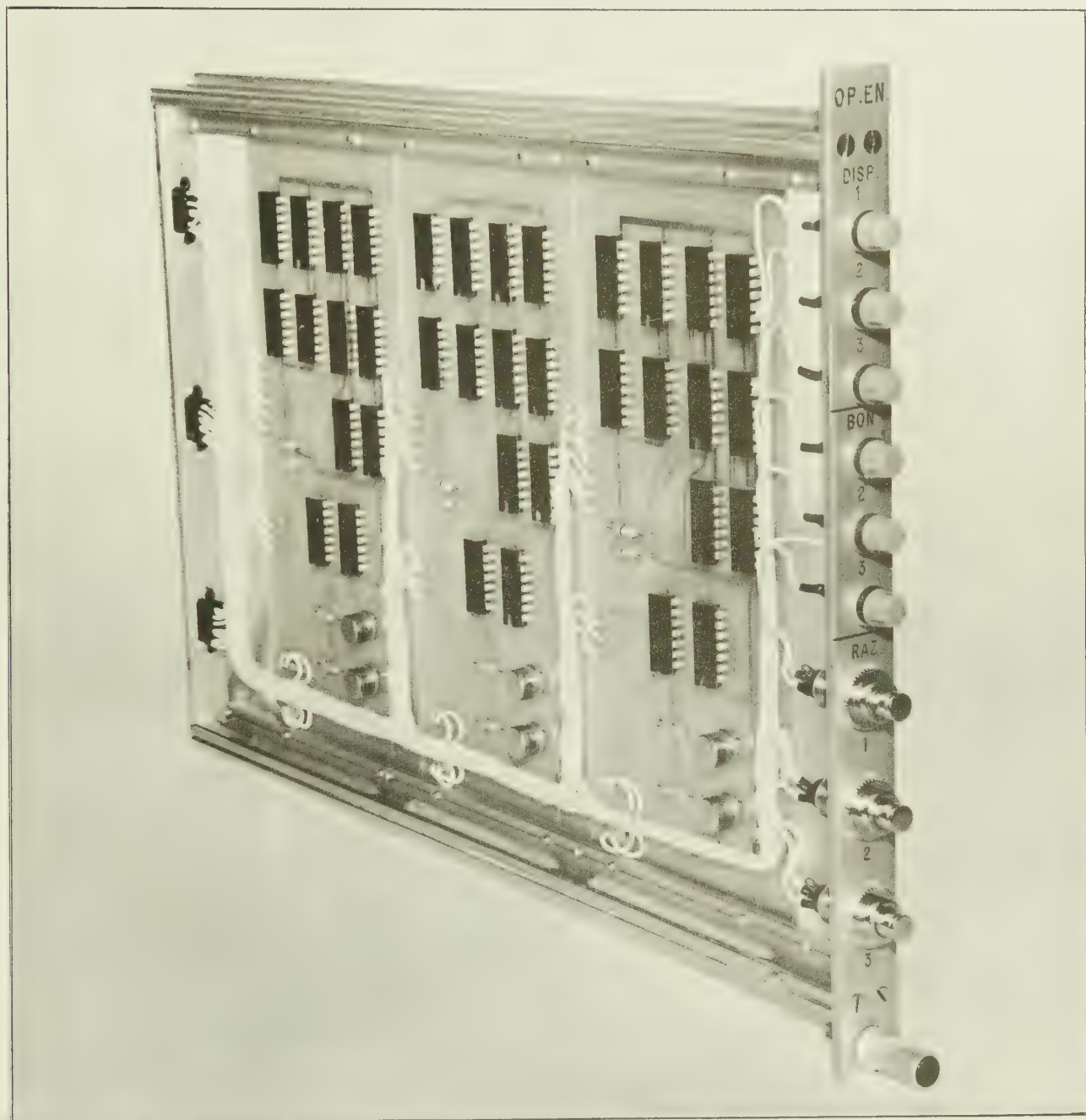


Fig. 5.14 - Vue avant du tiroir 1/25

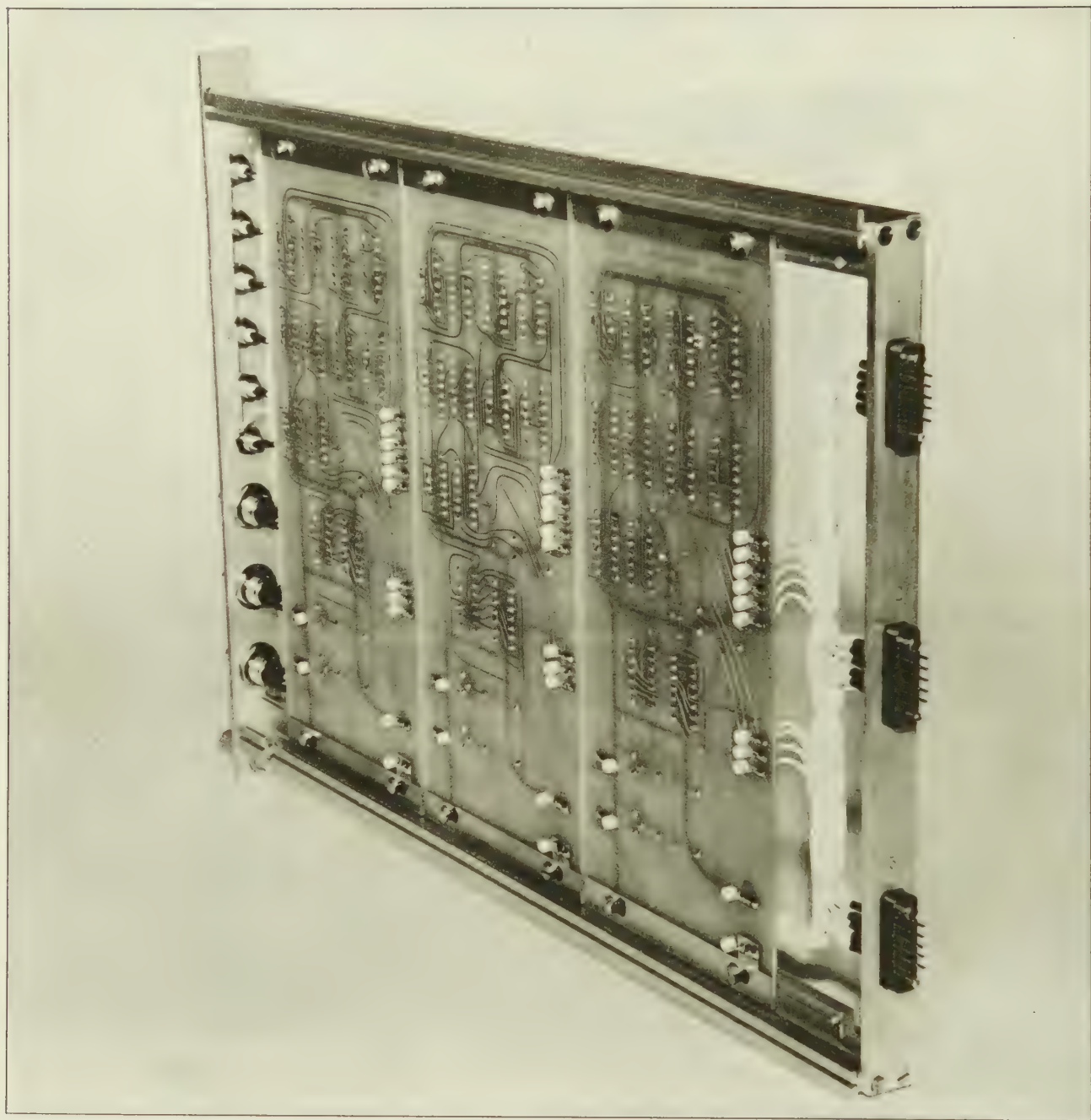


Fig. 5.15 - Vue arrière du tiroir 1/25



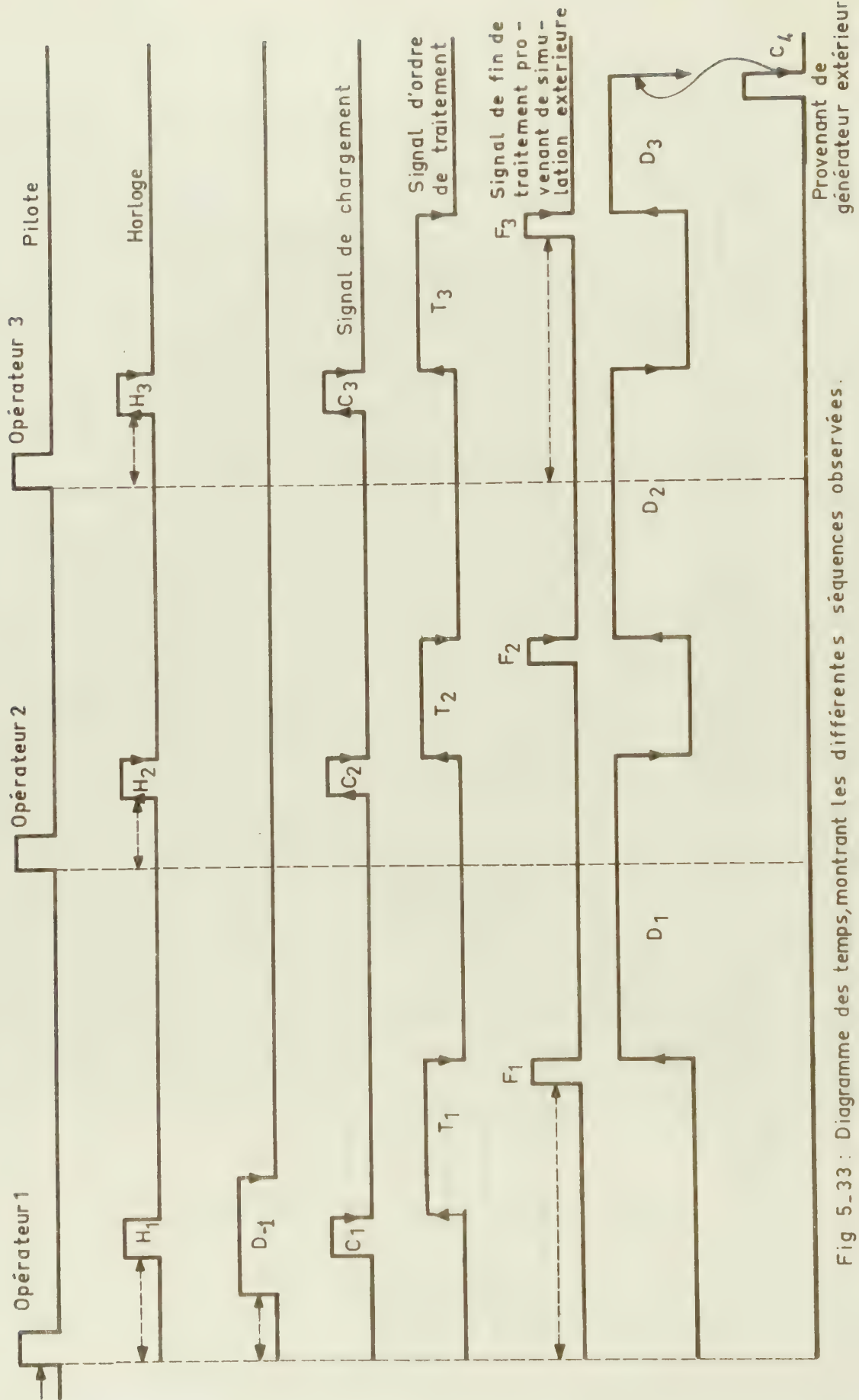


Fig 5.33 : Diagramme des temps, montrant les différentes séquences observées.

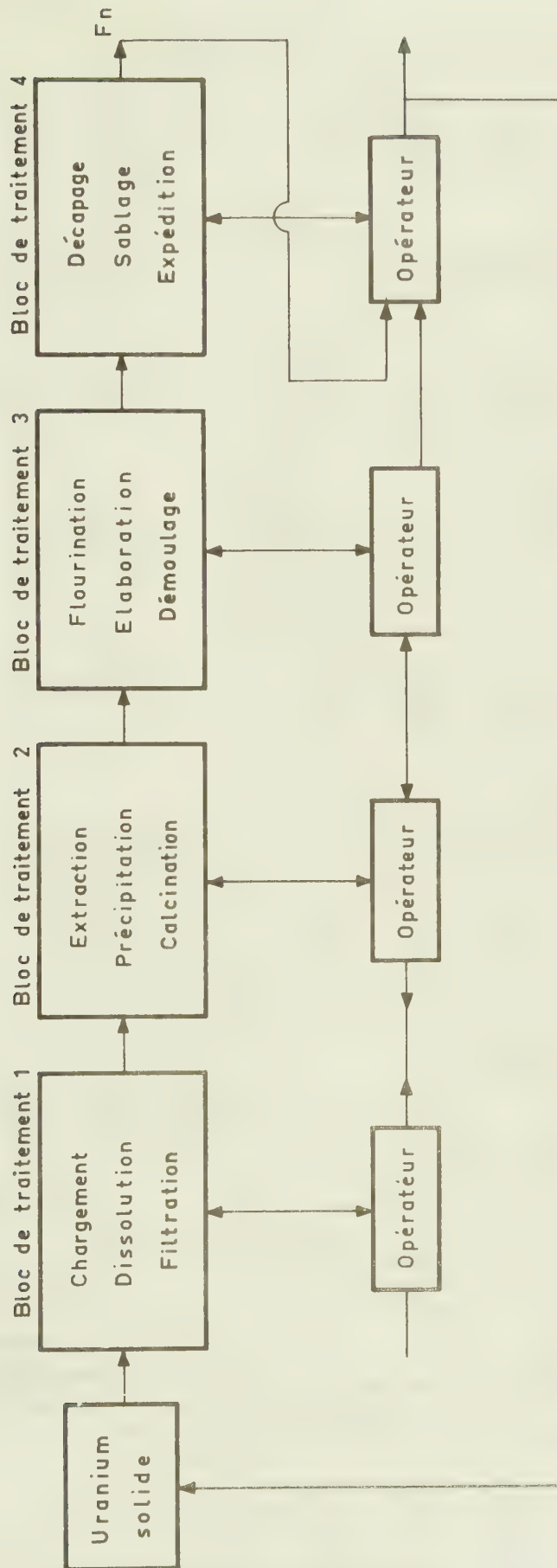


Fig. 5 - 34 : Schéma simplifié d'une chaîne de raffinage d'uranium pouvant être contrôlée par des opérateurs d'enchaînement.

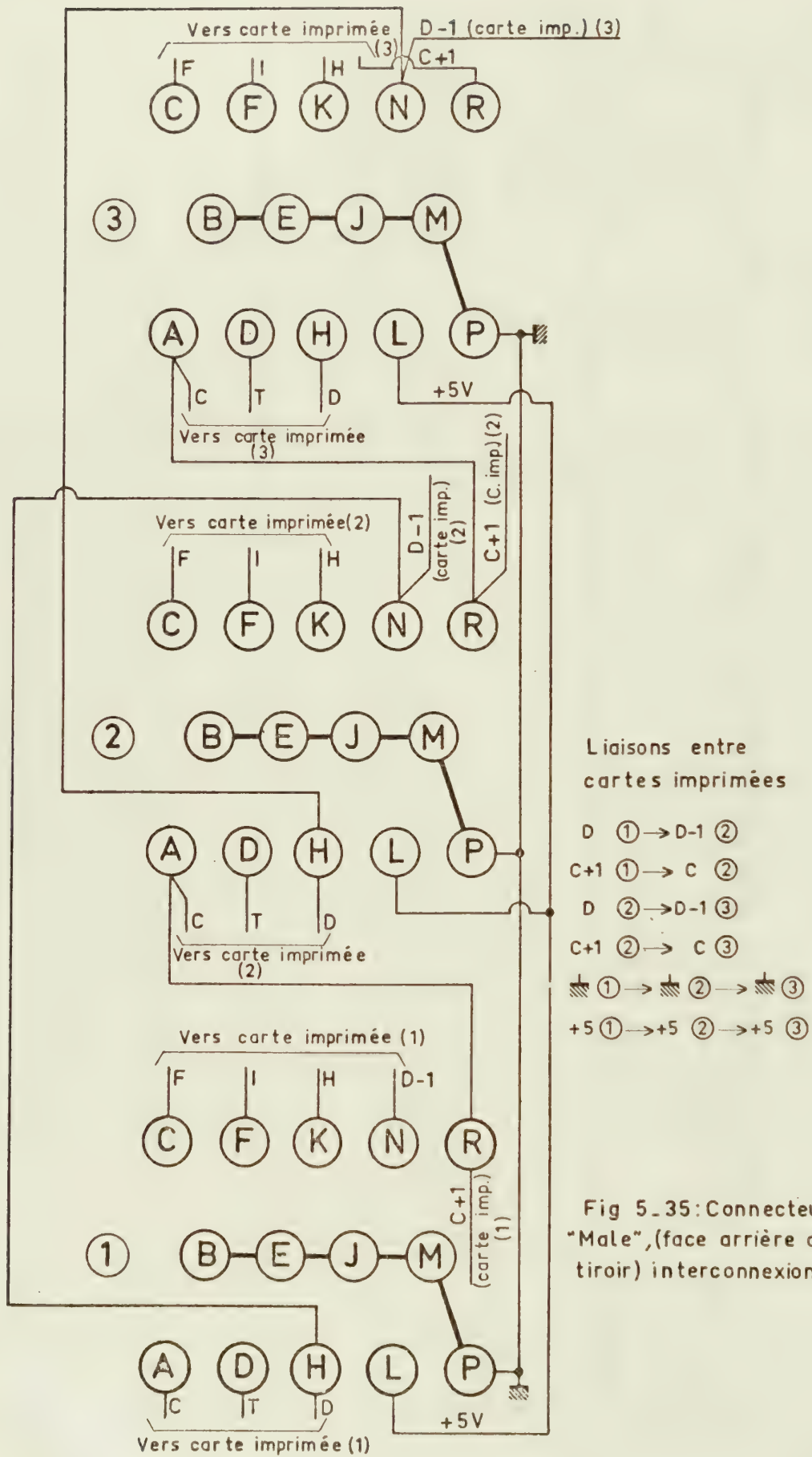


Fig 5.35: Connecteur "Male", (face arrière du tiroir) interconnexions.

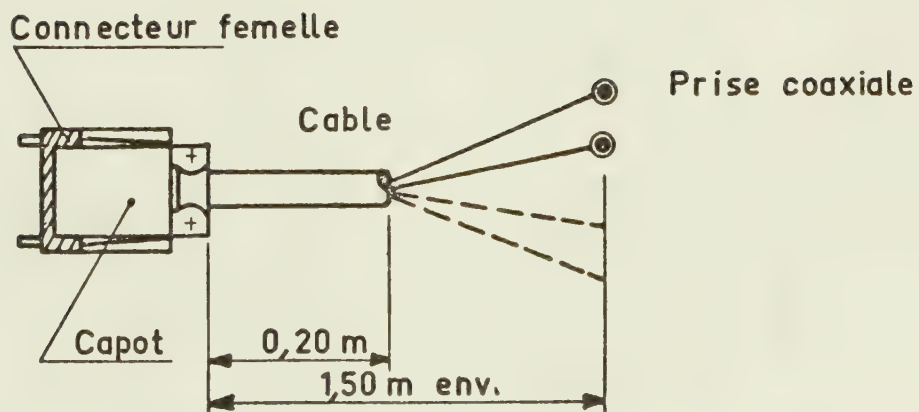
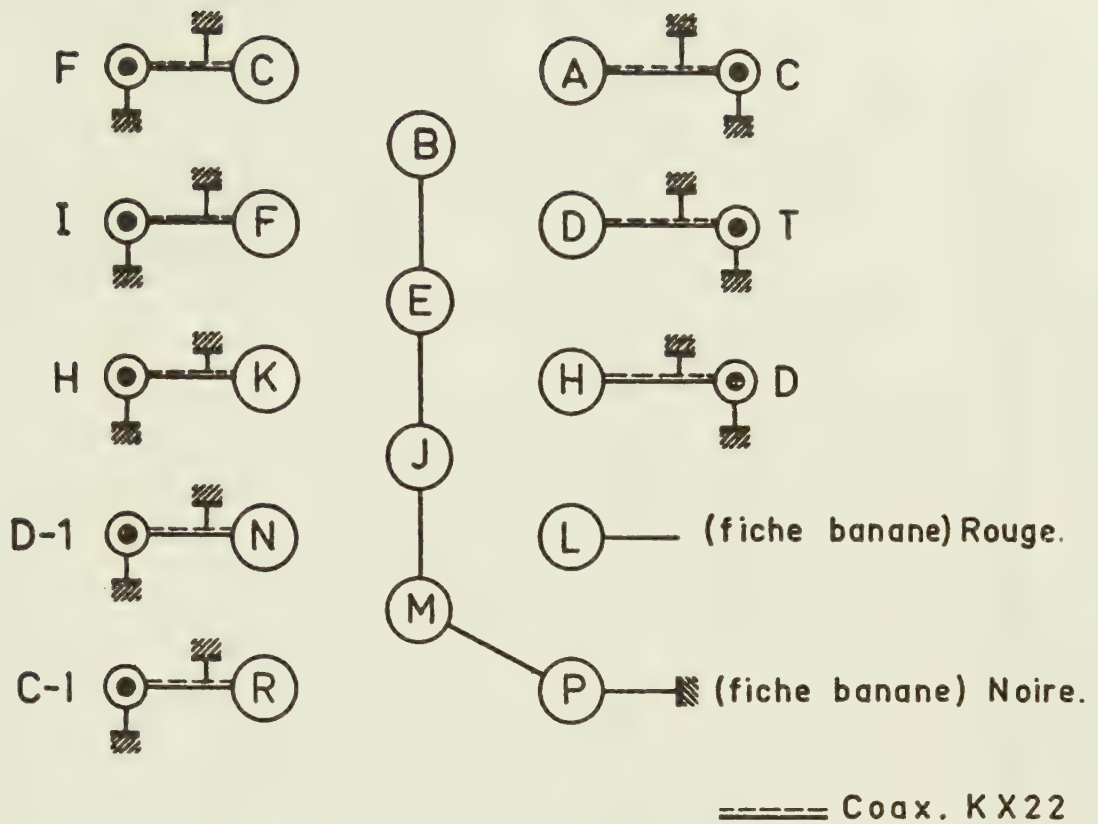


Fig 5\_36 : Connecteur femelle - Liaisons avec cable.



Fig 5\_37 : Face avant du tiroir 1/25

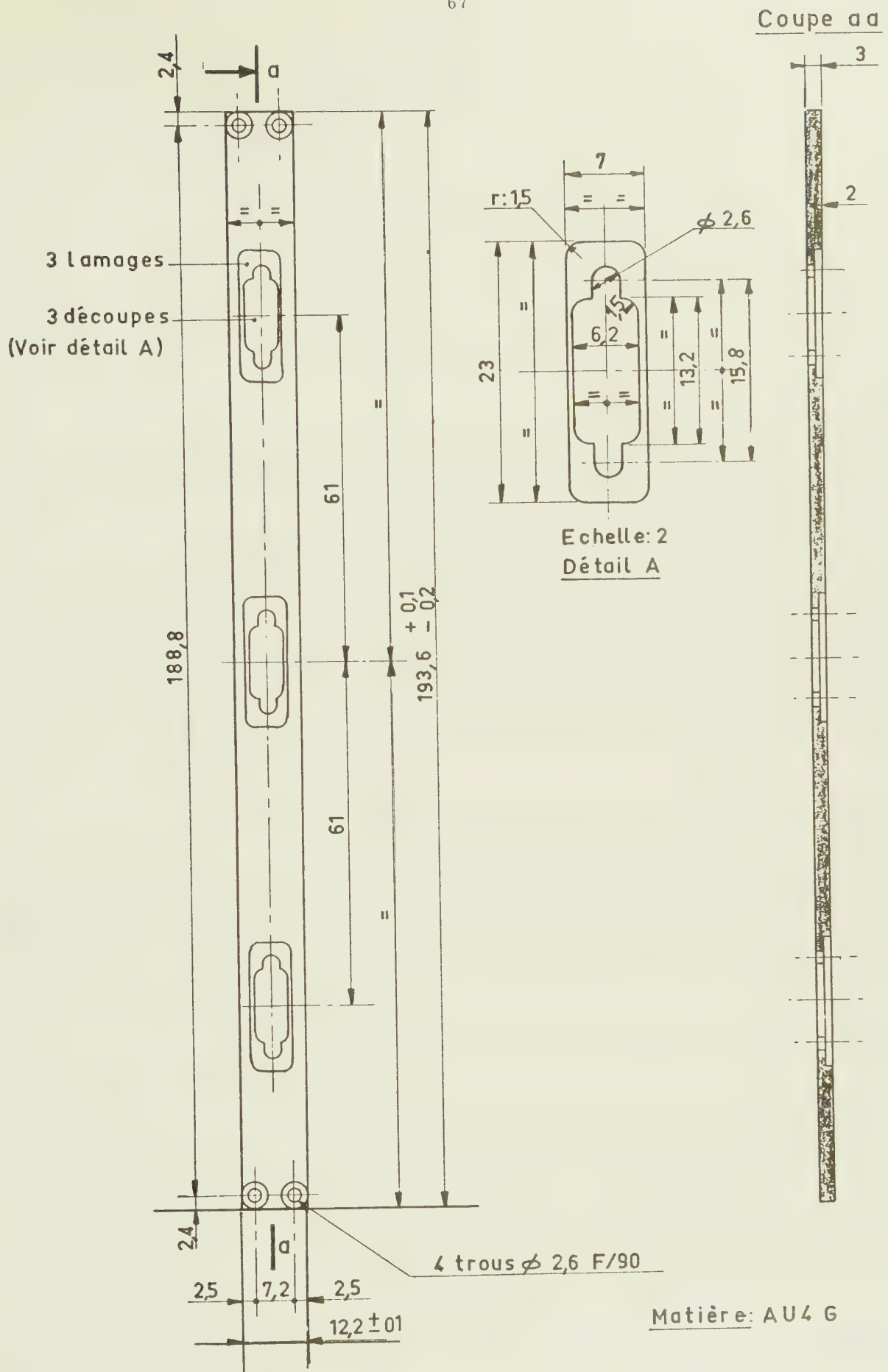


Fig 5 - 38: Face arrière du tiroir 1/25.



## CONCLUSION

La procédure adoptée pour la conception et la réalisation des opérateurs d'enchaînement peut être appliquée à n'importe quel autre système numérique, à condition de connaître les bases fondamentales et les méthodes graphiques de l'analyse binaire.

L'utilisation de graphes séquentiels particuliers qui indiquent les états commutants, les états commutés, les états de commutation et toutes les transitions, à l'opposé des méthodes employées jusqu'ici, se révèle parfaitement adaptée à l'étude et à la réalisation pratique des systèmes séquentiels en général. Nous disposons maintenant, véritablement, d'une méthode de synthèse rigoureuse, objective et pourvue, dans ses aspects théoriques, d'une grande souplesse d'adaptation à une technologie qui peut évoluer, fluctuer, mais à laquelle il est toujours possible d'associer une représentation mathématique précise en accord avec les principes et les définitions de l'analyse binaire.

C'est ainsi qu'une fonction dibinaire peut être réalisée dans le concret, en faisant appel à des moyens mécaniques, pneumatiques, électromécaniques aussi bien qu'électronique ou optoélectroniques, et que la technologie ne change rien à l'existence des quatre états consécutifs stables et aux impératifs de la condition des seuils qui entrent dans la définition de ces fonctions. Les aléas mêmes, qui sont longtemps restés la source de nombreuses difficultés et l'origine de nombreuses pannes, peuvent être localisés et éliminés grâce au théorème de non simultanéité, aux "consensus" et aux fonctions "délais".

Les différentes étapes qui ressortent de l'application de ces méthodes peuvent être rapidement résumées de la façon suivante :

1° - On traduit les données d'un problème en diagramme des temps et en implications essentielles, de manière à séparer au départ les fonctions génératrices, les fonctions de transcodage et les fonctions "réflexes". Chaque fonction est ensuite traitée séparément.

2° - On calcule les fonctions de transcodage en établissant leur table de vérité, et les fonctions génératrices en étudiant l'ordre chronologique des transitions engendrées. Les fonctions "réflexes", pour leur part, conduisent au tracé d'un graphe des fonctions primitives.

3° - L'examen de la condition des états permet de tracer, à partir du graphe des fonctions primitives, un graphe réflexe optimal.

4° - Chaque composante binaire de la fonction "réflexe" correspondant au graphe réflexe optimal, peut être calculée en repérant sur le graphe les points commutés et commutants relatifs à cette composante. On obtient ainsi pour chaque fonction "réflexe" élémentaire "mémoire ou binaire" un graphe d'émergence qui la caractérise totalement et qui permet de calculer son expression mathématique.

Pour obtenir la fonction séquentielle cherchée il faut compléter éventuellement par le calcul de la fonction de transcodage dans laquelle la fonction réflexe intervient comme vecteur variable.

5° - Il faut ensuite choisir la technologie et, en fonction de celle-ci, procéder à la localisation des aléas, en recherchant les fonctions carrées biformes et en appliquant à toutes les transitions le théorème de non simultanéité.

L'introduction de consensus redondants et de fonctions génératrices permet d'éliminer les aléas de continuité et de simultanéité gênants, ou bien de masquer, dans certains cas, les aléas de frontière.

6° - Le travail d'étude théorique étant achevé, on peut, en partant des expressions mathématiques, tracer les logigrammes et procéder à la construction du système qui répond au problème posé.

Sur un plan très général, un ensemble de traitement numérique de l'information peut être réalisé de très nombreuses manières différentes.

C'est pourquoi il est nécessaire de faire une analyse préalable des possibilités de décomposition de chaque système complexe.

Ces possibilités de décomposition dépendent, tout à la fois, de l'état de nos connaissances et du développement de la technologie.

Nous possédons actuellement un premier critère de décomposition qui consiste à distinguer, dans un système, les parties ayant un caractère synchrone de celles dont le caractère est inévitablement asynchrone. C'est ainsi, que parmi les trois composantes binaires du vecteur fonction de l'opérateur d'enchaînement, nous avons dû séparer le sous-système "C" du sous-système "T, D".

Ce critère procède de la différence qui existe toujours entre "prévu" et "imprévisible".

Les opérateurs d'enchaînement sont indispensables à la coordination des processus ayant un caractère aléatoire partiel et peuvent intervenir avec une rapidité qui dépend des caractéristiques des circuits utilisés.

Ces caractéristiques sont telles qu'elles assurent actuellement une très grande fiabilité d'intervention relativement au temps de déroulement des processus eux-mêmes.

Ces opérateurs sont des microprogrammes de contrôle modulaires câblés dont la souplesse et l'autonomie permettent leur utilisation, non seulement dans des systèmes de traitement de l'information décentralisés, mais aussi dans tous les systèmes automatiques à fonc-

tionnement numérique. Les opérateurs d'enchaînement peuvent, dès maintenant, être utilisés, aussi bien pour le contrôle d'une installation de raffinage de l'uranium (fig. 5.34), que pour la commande d'une raffinerie de pétrole. Ils peuvent commander et contrôler tout autre système du même genre et les mêmes solutions sont utilisables pour l'étude et la réalisation de systèmes de signalisation et de contrôle routiers ou ferroviaires.

Bien que, seules, trois opérations : l'acquisition, le traitement et la mise à disposition des résultats, aient été envisagées dans ce cas, les opérateurs d'enchaînement constituent la base essentielle à partir de laquelle peuvent être élaborés des opérateurs programmes plus complexes susceptibles d'apporter des solutions valables et décisives au développement des systèmes de l'informatique.



## BIBLIOGRAPHIE

- [1] VALLEE R.L.  
De l'intérêt d'une organisation locale des structures dans les automates numériques.  
Bulletin d'Instrumentation Nucléaire, n° 36, Mars 1969.
- [2] VALLEE R.L.  
Analyse binaire  
Tome I, Masson 1970.
- [3] VALLEE R.L.  
Analyse binaire  
Tome II, Masson 1970.
- [4] GOLENKO D.I.  
Computer Elements and Systems (Translated from russian) page 166, 1964.
- [5] YU A. AVDEEV and NIKOLAEVA A.P.  
Project control using the critical path Technique, Computer  
Elements and Systems (translated from russian) page 92, 1964.
- [6] VERGEZ P.  
Analyse du fonctionnement logique de la bascule TTL 5472  
Note CEA-N 1237, Janvier 1970.
- [7] DAVIS R.L.  
The Illiac IV Processing Element  
I.E.E.E. Transactions on Computers vol. C-18, number 9, Sept. 1969.
- [8] STOCKES R.  
Illiac IV : Route to parallel Computers  
Electronic Design 26, December 1967.
- [9] GREGORY J. and Mc REYNOLDS R.  
The Solomon Computer  
I.E.E.E. Transactions on Electronic Computers, December 1963.
- [10] KOCZLA L.J. and WANG G.Y.  
I.E.E.E. Transaction on Computers, vol. C-18 n° 6, June 1969.
- [11] HOLLAND J.  
A universal machine capable of executing an arbitrary number of sub programs simulta-  
neously.  
Proc. Eastern joint Computer Conf. New York, 1959.

- [12] CAMPEAU J.O.  
The block oriented Computer  
I.E.E.E. Transaction on Computers, vol. C-18, n° 8, August 1969.
- [13] European Atomic Energy Community (Euratom)  
"CAMAC" - A modular instrumentation system for data handling.
- [14] VALLEE R.L.  
Analyse binaire, 1ère partie, rapport CEA-R 3534.

Vu et approuvé  
ORSAY, le 2 juillet 1970  
Le Doyen de la Faculté des Sciences  
d'Orsay  
G. POITOU





